



## OpenEdge Management and OpenEdge Explorer: Configuration

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

A (and design), Actional, Actional (and design), Allegrix, Allegrix (and design), Apama, Apama (and Design), Business Empowerment, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Technologies, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, EasyAsk, EdgeXtend, Empowerment Center, Fathom, IntelliStream, Neon, Neon New Era of Networks, O (and design), ObjectStore, OpenEdge, PeerDirect, Persistence, POSSENET, Powered by Progress, PowerTier, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, Progress Sonic, ProVision, PS Select, SequeLink, Shadow, ShadowDirect, Shadow Interface, Shadow Web Interface, SOAPStation, Sonic, Sonic ESB, SonicMQ, Sonic Orchestration Server, Sonic Software (and design), SonicSynergy, SpeedScript, Stylus Studio, Technical Empowerment, 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 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, Who Makes Progress, and Your World. Your SOA. are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Any other trademarks or trade names 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](mailto:apache@apache.org). Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge Management includes software developed by Sun Microsystems, Inc. Copyright © 2003 Sun Microsystems, Inc. All Rights Reserved. and 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 Licenses for the specific language governing rights and limitations under the License agreement that accompanies each 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 include software developed by the ModelObjectsGroup (<http://www.modelobjects.com>). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from the SSC Software without prior written permission. Products derived from the SSC Software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact [djacobs@modelobjects.com](mailto:djacobs@modelobjects.com).

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..



---

# Contents

---

<b>Preface</b> .....	<b>Preface–1</b>
<b>1. Introduction</b> .....	<b>1–1</b>
OpenEdge Management and OpenEdge Explorer .....	1–2
OpenEdge Management .....	1–2
OpenEdge Explorer .....	1–3
Getting started .....	1–4
Comparing OpenEdge Management and OpenEdge Explorer .....	1–4
<b>2. Configuring OpenEdge Databases</b> .....	<b>2–1</b>
Database configuration and administration .....	2–2
Viewing the database log file .....	2–3
Database configuration awareness .....	2–3
Managing databases .....	2–4
Viewing or modifying database properties .....	2–5
Viewing the various database configuration properties .....	2–6
Editing the database default configuration .....	2–13
Creating a database configuration .....	2–14
Assigning a different default database configuration .....	2–15
Viewing the server group .....	2–16
Editing a database server group .....	2–19
Creating a database server group .....	2–19
Starting or stopping the database .....	2–20
Starting a database automatically .....	2–21
Using the background writers and watchdog process .....	2–22
Background writers .....	2–22
The watchdog process .....	2–22
Viewing database status .....	2–23
Viewing database process details .....	2–23
Deleting a database, database configuration, or database server group .....	2–25
Deleting a database .....	2–25
Deleting a database configuration .....	2–25
Deleting a database server group .....	2–26

Configuring an OpenEdge Replication-enabled database .....	2-27
Rules for setting properties for a Replication-enabled database .....	2-27
Setting Replication-related properties .....	2-28
Setting OpenEdge Replication Agent properties .....	2-28
Setting OpenEdge Replication Server properties .....	2-30
Setting OpenEdge Replication Control Agent properties .....	2-32
Setting OpenEdge Replication Transition properties .....	2-33
<b>3. Configuring DataServers .....</b>	<b>3-1</b>
ODBC DataServer configuration and administration .....	3-2
Working with the ODBC DataServer .....	3-2
Configuring an ODBC DataServer broker .....	3-3
Editing an ODBC DataServer broker configuration .....	3-8
Creating an ODBC DataServer broker .....	3-9
Deleting an ODBC DataServer broker .....	3-9
Starting an ODBC DataServer broker .....	3-9
Starting an ODBC DataServer broker automatically .....	3-10
Stopping an ODBC DataServer broker .....	3-10
Viewing the status of an ODBC DataServer broker .....	3-10
Viewing the ODBC DataServer log files .....	3-12
Oracle DataServer configuration and administration .....	3-14
Working with the Oracle DataServer broker .....	3-14
Configuring an Oracle DataServer broker .....	3-15
Editing an Oracle DataServer broker configuration .....	3-20
Creating an Oracle DataServer broker .....	3-20
Deleting an Oracle DataServer broker .....	3-21
Starting an Oracle DataServer broker .....	3-21
Starting an Oracle DataServer broker automatically .....	3-21
Stopping an Oracle DataServer broker .....	3-22
Viewing the status of an Oracle DataServer broker .....	3-22
Viewing the Oracle DataServer broker log files .....	3-24
MS SQL Server DataServer configuration and administration .....	3-26
Working with the MS SQL Server DataServer broker .....	3-26
Configuring an MS SQL Server DataServer broker .....	3-27
Editing an MS SQL Server DataServer broker configuration .....	3-32
Creating an MS SQL Server DataServer broker .....	3-33
Deleting an MS SQL Server DataServer broker .....	3-33
Starting an MS SQL Server DataServer broker .....	3-34
Starting an MS SQL Server DataServer broker automatically .....	3-34
Stopping an MS SQL Server DataServer broker .....	3-34
Viewing the status of an MS SQL Server DataServer broker .....	3-35
Viewing the MS SQL Server DataServer log files .....	3-36
<b>4. Configuring NameServers .....</b>	<b>4-1</b>
NameServer configuration and administration .....	4-2
NameServer features .....	4-2
Working with the NameServer .....	4-3
NameServer as an optional client connection request handler .....	4-3
Local and remote NameServers .....	4-3
Configuring a NameServer .....	4-4
Configuring local instances .....	4-4
Configuring remote instances .....	4-9
Editing a NameServer configuration .....	4-10
Setting or deleting NameServer environment variables .....	4-10
Creating a NameServer .....	4-12
Deleting a NameServer .....	4-13

Configuring fault tolerance and load balancing .....	4-14
Connection-level fault tolerance .....	4-14
Server-level fault tolerance .....	4-14
Load balancing .....	4-15
Location transparency .....	4-15
Starting a NameServer .....	4-16
Starting a NameServer automatically .....	4-16
Stopping a NameServer .....	4-17
Viewing the status of a NameServer .....	4-18
Handling UDP and firewall issues .....	4-20
Viewing the NameServer log file .....	4-21
 <b>5. Configuring AppServers and AppServer Internet Adapters .....</b>	<b>5-1</b>
AppServer configuration and administration .....	5-2
Understanding AppServer configuration .....	5-2
AppServer features .....	5-2
Working with the AppServer .....	5-3
AppServer configuration properties .....	5-3
Configuring an AppServer .....	5-4
Editing an AppServer configuration .....	5-21
Creating an AppServer .....	5-22
Starting and stopping an AppServer .....	5-22
Adding or trimming AppServer agents .....	5-23
Viewing the status of an AppServer .....	5-25
Deleting an AppServer .....	5-28
Viewing the AppServer log files .....	5-28
For additional AppServer information .....	5-29
AppServer Internet Adapter configuration and administration .....	5-30
Working with the AppServer Internet Adapter .....	5-30
Configuring an AppServer Internet Adapter .....	5-31
Editing an AppServer Internet Adapter configuration .....	5-37
Creating an AppServer Internet Adapter .....	5-37
Viewing the AppServer Internet Adapter log file .....	5-38
Enabling or disabling an AppServer Internet Adapter broker .....	5-39
Deleting an AppServer Internet Adapter broker .....	5-40
 <b>6. Configuring OpenEdge Web Services .....</b>	<b>6-1</b>
Configuring and managing Web services .....	6-2
SOAP-AVM gateway .....	6-2
Web Service Application Server .....	6-2
Managing Web services with the WSA .....	6-3
Web service session models and the WSA .....	6-3
Managing Web Services Adapter instances .....	6-5
Creating a new Web Services Adapter instance .....	6-5
Configuring a Web Services Adapter instance .....	6-7
Editing a Web Services Adapter instance configuration .....	6-16
Starting Web Services Adapter instances .....	6-17
Setting Web service defaults .....	6-17
Deploying a Web service to a Web Services Adapter instance .....	6-18
Exporting a Web service .....	6-20
Importing a Web service to a Web Services Adapter instance .....	6-20
Listing the deployed Web services for a Web Services Adapter instance .....	6-21
Viewing the status of a Web Services Adapter instance .....	6-22
Viewing Web Services Adapter instance statistics .....	6-22
Viewing statistics for deployed Web services .....	6-23
Changing Web Services Adapter Instance run-time properties .....	6-23

Stopping Web Services Adapter instances .....	6-24
Deleting a Web Services Adapter instance .....	6-24
Working with Web services .....	6-25
Enabling or disabling a Web service for client access .....	6-25
Viewing the status of a Web service .....	6-26
Viewing Web service statistics .....	6-26
Updating a deployed Web service .....	6-27
Undeploying a Web service .....	6-30
<b>7. Configuring WebSpeed Messengers and WebSpeed Transaction Servers ..</b>	<b>7-1</b>
WebSpeed Messenger administration .....	7-2
Configuring a WebSpeed Messenger .....	7-2
Editing a WebSpeed Messenger configuration .....	7-7
Viewing the WebSpeed Messenger log file .....	7-7
WebSpeed Transaction Server administration .....	7-9
Working with the WebSpeed Transaction Server .....	7-9
Configuring a WebSpeed Transaction Server .....	7-9
Editing a WebSpeed Transaction Server configuration .....	7-27
Creating a WebSpeed broker .....	7-28
Starting and stopping a WebSpeed broker .....	7-28
Adding or trimming WebSpeed agents .....	7-29
Deleting a WebSpeed broker .....	7-31
For additional WebSpeed information .....	7-32
<b>8. Configuring SonicMQ Adapters .....</b>	<b>8-1</b>
SonicMQ Adapter Broker configuration and administration .....	8-2
Working with the SonicMQ Adapter .....	8-3
Configuring a SonicMQ Adapter .....	8-4
Viewing or modifying SonicMQ Adapter broker properties .....	8-4
Viewing or modifying SonicMQ Adapter server properties .....	8-8
Viewing or modifying SonicMQ Adapter SSL properties .....	8-10
Setting or deleting SonicMQ Adapter environment variables .....	8-12
Enabling or disabling a SonicMQ Adapter broker .....	8-12
Editing a SonicMQ Adapter configuration .....	8-13
Creating a SonicMQ Adapter broker .....	8-14
Deleting a SonicMQ Adapter broker .....	8-15
Starting a SonicMQ Adapter .....	8-16
Stopping a SonicMQ Adapter .....	8-17
Viewing the status of a SonicMQ Adapter broker .....	8-18
Viewing the SonicMQ Adapter log file .....	8-20
<b>A. Reference to OpenEdge Web Service Properties .....</b>	<b>A-1</b>
Web services overview .....	A-2
Summary of properties .....	A-2
Setting service properties .....	A-4
Alphabetical reference .....	A-5
appServiceHost .....	A-5
appServiceName .....	A-5
appServicePort .....	A-5
appServiceProtocol .....	A-5
connectionLifetime .....	A-6
idleSessionTimeout .....	A-6
initialSessions .....	A-6
maxSessions .....	A-6
minIdleConnections .....	A-7
minSessions .....	A-7

noHostVerify .....	A-7
noSessionReuse .....	A-7
nsClientMaxPort .....	A-8
nsClientMinPort .....	A-8
nsClientPicklistExpiration .....	A-8
nsClientPicklistSize .....	A-8
nsClientPortRetry .....	A-9
nsClientPortRetryInterval .....	A-9
requestWaitTimeout .....	A-9
serviceFaultLevel .....	A-10
serviceLoggingLevel .....	A-10
staleO4GLObjectTimeout .....	A-10
waitIfBusy .....	A-11
<b>Index .....</b>	<b>Index-1</b>

**Tables**

Table 1–1:	OpenEdge Explorer functionality by resource . . . . .	1–3
Table 1–2:	OEM and OEE feature comparison . . . . .	1–5
Table 2–1:	Database properties . . . . .	2–5
Table 2–2:	General database configuration properties . . . . .	2–6
Table 2–3:	International database configuration properties . . . . .	2–8
Table 2–4:	Background writers database configuration properties . . . . .	2–9
Table 2–5:	Advanced database configuration properties . . . . .	2–11
Table 2–6:	Agent database configuration property . . . . .	2–13
Table 2–7:	General database server group properties . . . . .	2–16
Table 2–8:	Advanced database server group properties . . . . .	2–18
Table 2–9:	Process statistics data . . . . .	2–24
Table 3–1:	ODBC DataServer broker properties . . . . .	3–3
Table 3–2:	ODBC DataServer server properties . . . . .	3–7
Table 3–3:	ODBC DataServer broker status summary . . . . .	3–11
Table 3–4:	ODBC DataServer broker status details . . . . .	3–12
Table 3–5:	Oracle DataServer broker properties . . . . .	3–15
Table 3–6:	Oracle DataServer server properties . . . . .	3–19
Table 3–7:	Oracle DataServer broker property status summary . . . . .	3–22
Table 3–8:	Oracle DataServer broker status details . . . . .	3–23
Table 3–9:	MS SQL Server DataServer broker properties . . . . .	3–27
Table 3–10:	MS SQL Server DataServer server properties . . . . .	3–31
Table 3–11:	MS SQL Server DataServer broker property status summary . . . . .	3–35
Table 3–12:	MS SQL Server DataServer broker status details . . . . .	3–36
Table 4–1:	NameServer properties . . . . .	4–5
Table 4–2:	NameServer status summary properties . . . . .	4–18
Table 4–3:	NameServer status detail properties . . . . .	4–19
Table 5–1:	Ubroker.properties file structure . . . . .	5–3
Table 5–2:	AppServer Broker General properties . . . . .	5–5
Table 5–3:	AppServer Broker Owner Information properties . . . . .	5–5
Table 5–4:	AppServer Broker Controlling NameServer properties . . . . .	5–5
Table 5–5:	AppServer Broker AppService Name List properties . . . . .	5–6
Table 5–6:	AppServer Broker Logging Setting properties . . . . .	5–7
Table 5–7:	AppServer Broker Advanced Features properties . . . . .	5–10
Table 5–8:	AppServer Agent General properties . . . . .	5–12
Table 5–9:	AppServer Agent Logging Setting properties . . . . .	5–13
Table 5–10:	AppServer Agent Pool Range properties . . . . .	5–16
Table 5–11:	AppServer Agent Advanced Features properties . . . . .	5–16
Table 5–12:	AppServer SSL General properties . . . . .	5–18
Table 5–13:	AppServer SSL Advanced Features properties . . . . .	5–18
Table 5–14:	AppServer Messaging properties . . . . .	5–19
Table 5–15:	AppServer Internet Adapter General properties . . . . .	5–32
Table 5–16:	AppServer Internet Adapter Controlling NameServer properties . . . . .	5–32
Table 5–17:	AppServer Internet Adapter Logging Setting properties . . . . .	5–34
Table 5–18:	AppServer Internet Adapter SSL properties . . . . .	5–36
Table 5–19:	AppServer Internet Adapter Advanced Features properties . . . . .	5–37
Table 6–1:	Web Services Adapter Location properties . . . . .	6–9
Table 6–2:	Web Services Adapter Proxy Server Setting properties . . . . .	6–9
Table 6–3:	Web Services Adapter WSDL properties . . . . .	6–10
Table 6–4:	Web Services Adapter Logging Setting properties . . . . .	6–11
Table 6–5:	Web Services Adapter Security properties . . . . .	6–13
Table 6–6:	Web Services Adapter Advanced Features properties . . . . .	6–14
Table 7–1:	WebSpeed Messenger General properties . . . . .	7–3
Table 7–2:	WebSpeed Messenger Controlling NameServer properties . . . . .	7–3
Table 7–3:	WebSpeed Messenger Logging Setting properties . . . . .	7–5



---

Table 7-4:	WebSpeed Messenger SSL properties .....	7-6
Table 7-5:	WebSpeed Messenger Advanced properties .....	7-6
Table 7-6:	WebSpeed Broker General properties .....	7-10
Table 7-7:	WebSpeed Broker Owner Information properties .....	7-10
Table 7-8:	WebSpeed Broker Controlling NameServer properties .....	7-11
Table 7-9:	WebSpeed Broker AppService Name List properties .....	7-11
Table 7-10:	WebSpeed Broker Logging Setting properties .....	7-12
Table 7-11:	WebSpeed Broker Advanced Features properties .....	7-15
Table 7-12:	WebSpeed Agent General properties .....	7-17
Table 7-13:	WebSpeed Agent Logging Setting properties .....	7-18
Table 7-14:	WebSpeed Agent Pool Range properties .....	7-21
Table 7-15:	WebSpeed Agent Advanced Features properties .....	7-22
Table 7-16:	WebSpeed SSL General properties .....	7-24
Table 7-17:	WebSpeed SSL Advanced Features properties .....	7-24
Table 7-18:	WebSpeed Messaging properties .....	7-25
Table 8-1:	SonicMQ Adapter Broker General properties .....	8-4
Table 8-2:	SonicMQ Adapter Broker Owner Information properties .....	8-5
Table 8-3:	SonicMQ Adapter Broker Controlling NameServer properties .....	8-5
Table 8-4:	SonicMQ Adapter Broker AppServer Name List properties .....	8-6
Table 8-5:	SonicMQ Adapter Broker Logging Setting properties .....	8-7
Table 8-6:	SonicMQ Adapter Broker Advanced Features properties .....	8-8
Table 8-7:	SonicMQ Adapter Server General property .....	8-9
Table 8-8:	SonicMQ Adapter Server Logging Settings properties .....	8-9
Table 8-9:	SonicMQ Adapter Server Pool Range properties .....	8-10
Table 8-10:	SonicMQ Adapter SSL General properties .....	8-11
Table 8-11:	SonicMQ Adapter SSL Advanced Features properties .....	8-11
Table A-1:	Service properties .....	A-2
Table A-2:	requestWaitTimeout property values .....	A-9
Table A-3:	serviceLoggingLevel property values .....	A-10



---

# Preface

---

This Preface contains the following sections:

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

## Purpose

This manual describes how to establish property and configuration settings for OpenEdge databases, DataServers (for ODBC, Oracle, and MS SQL Server), NameServers, AppServers, AppServer Internet Adapters, Web Services Adapters, WebSpeed® Transaction Servers and Messengers, and SonicMQ® Adapters in OpenEdge® Management and OpenEdge Explorer. In addition, the manual also provides detail about viewing status and log files.

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

## Audience

This manual is intended for OpenEdge Management and OpenEdge Explorer users as well as OpenEdge Management database and system administrators.

## Organization

### Chapter 1, “Introduction”

Introduces OpenEdge Management and OpenEdge Explorer and describes their features.

### Chapter 2, “Configuring OpenEdge Databases”

Describes how to establish and modify property and configuration settings for a database, database configuration, or database server group; add a database instance; start and stop a database; use the background writers and watchdog process; view status; view the log file; and delete a database.

### Chapter 3, “Configuring DataServers”

Describes how to establish and modify property and configuration settings for an ODBC, Oracle, or MS SQL Server DataServer; add a DataServer instance; start and stop a DataServer; view status; view the log files; and delete a DataServer.

### Chapter 4, “Configuring NameServers”

Describes how to establish and modify property and configuration settings for a NameServer; add a NameServer instance; start and stop a NameServer; work with fault tolerance and load balancing; view status; view the log files; and delete a NameServer.

### Chapter 5, “Configuring AppServers and AppServer Internet Adapters”

Describes how to establish and modify property and configuration settings for an AppServer or AppServer Internet Adapter; add an AppServer or AppServer Internet Adapter instance; add or trim AppServer agents; start and stop an AppServer; enable or disable an AppServer Internet Adapter; view status; view the log files; and delete an AppServer or AppServer Internet Adapter.

### Chapter 6, “Configuring OpenEdge Web Services”

Describes how to configure and manage Web services; manage Web Services Adapter instances; and work with Web services.

### Chapter 7, “Configuring WebSpeed Messengers and WebSpeed Transaction Servers”

Describes how to establish and modify property and configuration settings for a WebSpeed Transaction Server or WebSpeed Messenger; create or delete a WebSpeed Transaction Server instance; start and stop a WebSpeed Transaction Server; add or trim WebSpeed Transaction Server agents; view status; and view the log files.

### Chapter 8, “Configuring SonicMQ Adapters”

Describes how to establish and modify property and configuration settings for a SonicMQ Adapter; create or delete a SonicMQ Adapter instance; start and stop a SonicMQ Adapter; view status; and view the log file.

### Chapter A, “Reference to OpenEdge Web Service Properties”

Provides a list of properties you can set for a Web service.

## 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
<b>Bold</b>	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.
<b>SMALL, BOLD CAPITAL LETTERS</b>	Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, <b>GET</b> and <b>CTRL</b> .
<b>KEY1+KEY2</b>	A plus sign between key names indicates a <b>simultaneous</b> key sequence: you press and hold down the first key while pressing the second key. For example, <b>CTRL+X</b> .
<b>KEY1 KEY2</b>	A space between key names indicates a <b>sequential</b> key sequence: you press and release the first key, then press another key. For example, <b>ESCAPE H</b> .
<b>Syntax:</b>	
Fixed width	A fixed-width font is used in syntax statements, code examples, system output, and filenames.
<i>Fixed-width italics</i>	Fixed-width italics indicate variables in syntax statements.
<b><i>Fixed-width bold</i></b>	Fixed-width bold indicates variables with special emphasis.
UPPERCASE fixed width	Uppercase words are ABL keywords. Although these are always shown in uppercase, you can type them in either uppercase or lowercase in a procedure.
	This icon (three arrows) introduces a multi-step procedure.
	This icon (one arrow) introduces a single-step procedure.
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.

Convention	Description
	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**.



# Introduction

---

OpenEdge® Management is a system management center that provides visibility, analysis, and proactive monitoring of critical information assets. OpenEdge Management optimizes the availability and performance of OpenEdge-based applications through system monitoring, alerting, and automatic handling of corrective actions. OpenEdge Management empowers Progress Software customers to become more efficient, decrease the cost of managing the OpenEdge® environment, and ensure high availability and performance.

OpenEdge Explorer is a subset of OpenEdge Management. It provides the functionality currently available in Progress Explorer, but within the OpenEdge Management console in a Web browser. OpenEdge Explorer enables you to set configuration properties, to start or stop, and to view the status or log files for various OpenEdge resources. Unlike Progress Explorer, which runs only in Windows, OpenEdge Explorer is supported on all the UNIX platforms that support OpenEdge.

This chapter introduces:

- [OpenEdge Management and OpenEdge Explorer](#)
- [Getting started](#)

# OpenEdge Management and OpenEdge Explorer

You can use OpenEdge Management or OpenEdge Explorer to configure your licensed OpenEdge resources.

---

**Note:** The AdminServer referenced in the following sections is a background process that provides a common point of management for the resources managed by OpenEdge Management or OpenEdge Explorer. For more information on the AdminServer, see *OpenEdge Getting Started: Installation and Configuration* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site.

---

## OpenEdge Management

OpenEdge Management provides database administrators and systems operations managers with the performance tools and processes required to configure, monitor, diagnose, and manage the OpenEdge environment. This is all done from an easy-to-use graphical interface known as the *management console*, which runs in a Web browser.

Using OpenEdge Management, you can monitor the following resources running under a local AdminServer or under an AdminServer on a remote machine:

- OpenEdge databases (including those that are enabled for OpenEdge® Replication)
- System resources (CPU, disk, memory, file system)
- File resources
- OpenEdge servers, such as the AppServer, NameServer, WebSpeed® Transaction Server, and supported DataServers (ODBC, Oracle, and MS SQL Server)
- WebSpeed Messengers
- AppServer Internet Adapters, Web Services Adapters, and SonicMQ® Adapters
- TCP-based network services

OpenEdge Management's deep level of monitoring provides information and details about your environment, enabling you to proactively manage operations.

## OpenEdge Explorer

OpenEdge Explorer allows you to set and modify properties from within the OpenEdge Management console. The functionality that you can use for the various resources in OpenEdge Explorer is provided in [Table 1-1](#).

**Table 1-1: OpenEdge Explorer functionality by resource**

OpenEdge Resource instance	Set config. props?	Edit config. props?	Start?	Stop?	Delete ?	Create ?	View generic status?	View log file?
AppServer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AppServer Internet Adapter	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Database	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Messengers	Yes	Yes	No	No	No	No	Yes	Yes
MSS DataServer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
NameServer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ODBC DataServer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Oracle DataServer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SonicMQ Adapter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WebSpeed Transaction Server	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Web Services Adapter	Yes	Yes	No	No	Yes	Yes	Yes	Yes

## Getting started

This guide provides details about how to use OpenEdge Management and OpenEdge Explorer to configure OpenEdge resources.

For introductory information about using the OpenEdge Management and OpenEdge Explorer console, see [OpenEdge Management and OpenEdge Explorer: Getting Started](#).

## Comparing OpenEdge Management and OpenEdge Explorer

The OpenEdge Management product is installed and licensed separately from OpenEdge. If you did not purchase OpenEdge Management, its options are grayed out and are unavailable in the management console. If you purchase and install OpenEdge Management, OpenEdge Explorer is also installed.

You can also install OpenEdge Explorer provided you have a supported version of OpenEdge installed.

In general, the settings you can establish and the options you can access when using OpenEdge Management and OpenEdge Explorer are determined both by the platform you are using and the products that you are licensed to use. If you notice that a particular property setting is dimmed in the OpenEdge Explorer console and not available for modification, it is because that property has no meaning within your operating environment or your licensed configuration.

---

**Note:** The settings (properties) for most of the resources managed by OpenEdge Management and OpenEdge Explorer (except for OpenEdge databases) are stored in the `ubroker.properties` file installed with OpenEdge. These properties are stored as name-value pairs. The names of these properties shown in OpenEdge Management and OpenEdge Explorer are long forms of the short names for the same properties stored in the `ubroker.properties` file. For more information on this file, see *OpenEdge Getting Started: Installation and Configuration* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site. For information on property storage for OpenEdge databases, see the “[Database configuration and administration](#)” section on page 2–2.

---

### Feature comparison

As described in [OpenEdge Management and OpenEdge Explorer: Installation](#), it is possible to install:

- OpenEdge Explorer only
- OpenEdge Management (which contains the OpenEdge Explorer functionality), if you have purchased the product

[Table 1–2](#) provides a comparison between the functionality available to you if you have installed only OpenEdge Explorer and the functionality available to you if you have purchased and installed OpenEdge Management (with OpenEdge Explorer).



**Table 1–2: OEM and OEE feature comparison***(1 of 2)*

<b>Feature</b>	<b>Available in OpenEdge Explorer?</b>	<b>Available in OpenEdge Management (with OpenEdge Explorer)?</b>
Configuring or modifying default property settings	Yes	Yes
Starting/stopping resource instances (if applicable)	Yes	Yes
Viewing status (if applicable)	Yes	Yes
Viewing log files	Yes	Yes
Adding and configuring new resource instances	Yes	Yes
Supporting resource and log file monitoring for: <ul style="list-style-type: none"> <li>Databases</li> <li>DataServers</li> <li>AppServers</li> <li>NameServers</li> <li>WebSpeed Transaction Servers</li> <li>Web Speed messengers</li> <li>Network components (TCP, UDP, PING, HTTP)</li> <li>System components (CPU, disk, file system, memory)</li> <li>File components</li> <li>SonicMQ Adapters</li> <li>Web Services Adapters</li> </ul>	No	Yes
Supporting log file monitoring for: <ul style="list-style-type: none"> <li>AppServer Internet Adapters</li> <li>WebSpeed Messengers</li> </ul>	No	Yes
Creating jobs for supported resources	No	Yes
Creating reports for supported resources	No	Yes
Using Library functions, such as rule sets, for supported resources	No	Yes
Alerts for supported resources	No	Yes
Scheduling and polling for supported resources	No	Yes
Graphing for supported resources	No	Yes

**Table 1–2: OEM and OEE feature comparison***(2 of 2)*

<b>Feature</b>	<b>Available in OpenEdge Explorer?</b>	<b>Available in OpenEdge Management (with OpenEdge Explorer)?</b>
Trending for supported resources	No	Yes
SNMP support	No	Yes (with SNMP license)
The Dashboard, My Collections	No	Yes

## Configuring OpenEdge Databases

---

You can use OpenEdge Management and OpenEdge Explorer to configure and administer the database configuration of an existing OpenEdge RDBMS.

This chapter provides the following information about database configuration:

- [Database configuration and administration](#)
- [Managing databases](#)
- [Viewing or modifying database properties](#)
- [Viewing the various database configuration properties](#)
- [Viewing the server group](#)
- [Starting or stopping the database](#)
- [Using the background writers and watchdog process](#)
- [Viewing database status](#)
- [Deleting a database, database configuration, or database server group](#)
- [Configuring an OpenEdge Replication-enabled database](#)

## Database configuration and administration

Using OpenEdge Management or OpenEdge Explorer, you can set properties for a database, its various configurations, and its associated server groups. Any configuration settings that you establish for a managed database, database configuration, or database server group in OpenEdge Management or OpenEdge Explorer are automatically reflected in the OpenEdge database Configuration Manager property file (`conmgr.properties`), and vice versa.

---

**Note:** You cannot create a new physical database from within OpenEdge Management or OpenEdge Explorer. For more information about creating a database, see *OpenEdge Data Management: Database Administration* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site.

---

A database recognized and managed by the AdminServer is called a *managed database*. A database not managed by an AdminServer is called a *scripted database*. Scripted databases are administered outside the AdminServer using parameter files (`.pf`) and operating system-dependent scripts.

A managed OpenEdge database starts up by using the default configuration information stored in the `conmgr.properties` file. The file contains the property settings for all managed databases on the host where OpenEdge is installed.

The `conmgr.properties` file resides in the `properties` subdirectory of the OpenEdge install directory. Each database is represented in the file by the following groups:

- **Database** — The group that describes the physical database. There is one database group for each physical database.
- **Configuration** — One or more groups that describe the database arguments and server groups for an active instance of the database. Only one database configuration can be active at one time. The active configuration is the default configuration identified in the server group.
- **ServerGroup** — One or more groups that describe brokering behavior. The first server group within a configuration defines the primary broker and each subsequent group defines a secondary broker. The properties in a server group define the arguments that are specific to that primary or secondary broker.

You can edit the `conmgr.properties` file manually; any changes that you make to it are automatically reflected in OpenEdge Management or OpenEdge Explorer.

---

**Note:** Although you can make manual edits to the `conmgr.properties` file, Progress Software recommends that you use the `mergeprop` utility, OpenEdge Management, or OpenEdge Explorer to make property changes. For more information about the `mergeprop` utility, see *OpenEdge Getting Started: Installation and Configuration*.

---

From OpenEdge Management or OpenEdge Explorer, you can perform the following database-related actions:

- Add an existing database
- Create a new database configuration
- Edit an existing or new database configuration
- Use the default server group
- Create a new database server group
- Edit a new or existing database server group
- Start a database
- Start a database automatically
- Stop a database
- Use the background writers and watchdog processes
- View the status of a database
- Delete a database
- Delete a database configuration (provided there is more than one)
- Delete a database server group (provided there is more than one)
- Configure the properties of an OpenEdge Replication-enabled database

## Viewing the database log file

You can use the log file viewer to review the database log file. For more information, see the section about using the log file viewer in *OpenEdge Management: Database Management*.

## Database configuration awareness

You can use OpenEdge Management or OpenEdge Explorer to establish database settings and options. These settings and options are determined both by the platform you are using and the products that you are licensed to use. If a property is not applicable to the database, its setting is either ignored or, when appropriate, an error or warning is issued.

## Managing databases

You can use OpenEdge Management or OpenEdge Explorer to manage your database. You cannot physically create a database in the management console; however, you can create the definition of a database so that you can configure, start, stop, or view status for the primary broker and all auxiliary processes associated with that broker, and view status for the log file for the database.

Database configuration information is stored in the `conmgr.properties` file. For a complete list of how the database properties map to the command-line `.pf` arguments, see the `conmgr.properties.README`. You can locate the `README` file in the `<OpenEdge-install>\properties` directory.

A database recognized and managed by the AdminServer is called a *managed* database. A database not managed by an AdminServer is called a *scripted* database. Scripted databases are administered outside the AdminServer using parameter files (`.pf`) and operating system-dependent scripts. You can migrate a scripted database to a managed database.

For details about adding a managed or a scripted database, see [OpenEdge Management: Database Management](#).

## Viewing or modifying database properties

Each database definition has a set of properties. You can view and modify the properties if you want to establish new values.



### To view or modify the database properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.

From the **Database Configuration** page, you can do the following:

- View a read-only display of the database properties as described in [Table 2–1](#).
- Click **Edit** to modify the database properties. When you finish, click **Save**.
- Click the relevant **Configuration and Server Group** link to access the database default configuration properties or the database server group properties. For details about these properties, see the “[Viewing the various database configuration properties](#)” section on page 2–6 and the “[Viewing the server group](#)” section on page 2–16.
- Set OpenEdge Replication Server or Agent properties, if the database is replication-enabled or if an associated `<db-name>.rep1.properties` file exists in the physical database directory. For details, see the “[Configuring an OpenEdge Replication-enabled database](#)” section on page 2–27.

**Table 2–1: Database properties**

Property	Description
<b>Database name</b>	Identifies the full path to the database.
<b>Auto start</b>	Indicates whether the database starts automatically when the AdminServer starts.
<b>Cluster enabled</b>	Indicates whether you have already configured the database as cluster-enabled (by using the PROUTIL DESCRIBE command). This allows the database to register with the cluster manager at start time.
<b>Default configuration</b>	Provides the name of the default configuration that is used when the database starts.

## Viewing the various database configuration properties

OpenEdge Management and OpenEdge Explorer provide a predefined default database configuration. Once you add a database instance, you can use the property settings established in the default configuration, you can modify the default settings, or you can create and modify additional configurations.

Note that only one database default configuration can be selected for a database at a time.



### To view the database default configuration properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click **configuration.<dbname>.defaultconfiguration**. The **Database Configuration** page appears, displaying the default configuration properties in the following five categories: **General**, **International**, **Background Writers**, **Advanced**, and **Agent**. These properties are described in [Table 2–2](#) through [Table 2–6](#), which follow this procedure.

From the **Database Configuration** page, you can do the following:

- View a read-only display of the database default configuration properties as described in [Table 2–2](#) through [Table 2–6](#).
- Click **Edit** to modify the default database properties. For details, see the “[Editing the database default configuration](#)” section on page 2–13.
- Click the relevant link to access the associated database server group properties. For details about these properties, see the “[Viewing the server group](#)” section on page 2–16.

**Table 2–2: General database configuration properties**

(1 of 2)

Property	Description
<b>Blocks in database buffers</b>	The number of blocks in the database buffers. The optimum value for this property depends on your application. The default value is 0.
<b>Hash table entries</b>	The number of hash table entries to use for the buffer pool. The default value is 0. <b>Note:</b> Do not edit this property unless directed to do so by Progress Software Technical Support.



**Table 2–2: General database configuration properties**

(2 of 2)

Property	Description
<b>Lock table entries</b>	<p>The number of entries in the record locking table.</p> <p>If you specify a value that is not a multiple of 32, the value is rounded to the next highest multiple of 32. Each record that is accessed by any user takes one entry.</p> <p>The default value is 8192.</p>
<b>Max users</b>	<p>The maximum number of users on the system for the database.</p> <p>After the maximum number of users have connected to the database, additional user connect attempts are rejected.</p> <p>The default value is 20.</p>
<b>Max servers</b>	<p>The maximum number of remote client servers that can be started by the broker process.</p> <p>Use the <b>Max servers</b> property to limit the number of remote user servers that can be started on the system. The performance trade-off to consider is swapping overhead for many servers versus overloading (slowing down) a server with too many clients.</p> <p><b>Note:</b> This parameter pertains only to the OpenEdge® Workgroup RDBMS and the OpenEdge® Enterprise RDBMS. In addition, this parameter applies only to databases that are accessed from remote network nodes.</p>
<b>Direct I/O</b>	<p>Enables the Direct I/O property, which opens all files in unbuffered mode. Direct I/O enables the database to use an I/O technique that bypasses the operating system buffer pool and transfers data directly from a buffer to disk.</p> <p>This technique has some advantages over buffered reads and writes, such as avoiding the overhead of maintaining the operating system buffer pool, and eliminating competition for operating system buffers between the database and other programs. The operating system buffer-pool algorithms are designed for efficient sequential file access; the OpenEdge buffer-pool algorithms are more efficient for access to an OpenEdge RDBMS.</p> <p>The <b>Direct I/O</b> check box is cleared by default.</p>
<b>Other arguments</b>	Any other arguments for which there is no defined property available.

**Table 2–3: International database configuration properties**

Property	Description
<b>Collation table (-cpcoll)</b>	<p>A collation table in the convmap.cp file that you use for collation rules.</p> <p>The collation table you specify is used with the code page OpenEdge uses in memory, which is specified by the Internal Code Page (-cpinternal) parameter.</p> <p>The default value is BASIC.</p>
<b>Case table name (-cpcase)</b>	<p>A case table in the convmap.cp file that you use for case rules.</p> <p>Case rules are used by the 4GL CAPS and LC functions and by the exclamation point ( ! ) formatting character, which you use to tell OpenEdge to convert all characters to uppercase during input. The case table you specify is used to establish case rules for the code page used in OpenEdge's memory. The memory code page is set with the Internal Code Page (-cpinternal) parameter.</p> <p>The default value is BASIC.</p>
<b>Conversion map (-convmap)</b>	<p>The conversion map file that OpenEdge uses for code page conversions, collation orders, and case conversions.</p> <p>By default, OpenEdge uses <i>OpenEdge-install-dir\convmap.cp</i> in Windows and <i>OpenEdge-install-dir/convmap.cp</i> on UNIX.</p>
<b>Server code page (-cpinternal)</b>	<p>The code page that OpenEdge uses in memory.</p> <p>For GUI clients, the -cpinternal code page should be the same code page the operating system uses.</p> <p>If you do not specify a value, OpenEdge uses the iso8859-1 code page by default.</p>
<b>Log code page (-cplog)</b>	<p>The code page that the database uses when it writes to a log file. By default, the database uses the internal code page.</p> <p>Choose a code page that agrees with your operating system and its devices. If you have a Unicode database, you might want to specify the log character set as Unicode so information is properly logged.</p> <p>If you do not specify a value, OpenEdge uses the iso8859-1 code page by default.</p>

**Table 2–4: Background writers database configuration properties** (1 of 2)

Property	Description
<b>Watchdog process</b>	<p>Start the Watchdog process automatically when the database starts.</p> <p>The check box is selected by default.</p>
<b>Asynchronous page writers</b>	<p>The number of Asynchronous Page Writer (APW) processes to start.</p> <p>A database can have between zero and nine APWs running simultaneously. The optimal number is dependent on your application and environment.</p> <p>To start, use one page writer for each disk where the database resides. If this is insufficient, add more.</p> <p>For an application that performs many updates, start one APW for each disk containing your database, and add one additional APW.</p> <p>Applications that perform fewer changes to a database require fewer APWs. If an application performs no updates, no page writers are required.</p> <p>The default value is 1.</p>
<b>Before image process</b>	<p>Start a Before-imaging process automatically when the database starts.</p> <p>The check box is selected by default.</p>
<b>Before image stall</b>	<p>Send a message to the log file when the recovery log threshold is reached, preventing emergency shutdown. (See the documentation on the <code>-bi stall</code> startup parameter on PSDN at <a href="http://www.psdn.com/library/kbcategory.jspa?categoryID=129">http://www.psdn.com/library/kbcategory.jspa?categoryID=129</a>.)</p> <p>The check box is cleared by default.</p>
<b>Before image buffers</b>	<p>The number of before-image buffers to use.</p> <p>This property is useful only when running the before-image writer (BIW). The BIW continually writes the filled before-image buffers to disk, making the buffers available to other client and server processes. Without a BIW writing the buffers, any gain from increasing the number of buffers is negligible.</p> <p>The default value is 5.</p>
<b>Before image threshold</b>	<p>The maximum size to which BI files can grow as a percentage of the largest possible recovery log file. (See the documentation on the <code>-bi thold</code> startup parameter on PSDN at <a href="http://www.psdn.com/library/kbcategory.jspa?categoryID=129">http://www.psdn.com/library/kbcategory.jspa?categoryID=129</a>.)</p> <p>The default value is 3.</p>

**Table 2–4: Background writers database configuration properties** (2 of 2)

Property	Description
<b>Before image delay writes</b>	<p>A positive value that you specify to delay OpenEdge from synchronously writing out to disk the last before-image (BI) file records at the end of each transaction.</p> <p>On UNIX systems using shared memory, the value also specifies the interval at which the broker process wakes up to make sure all BI file changes have been written to disk.</p> <p>The default value is 3.</p>
<b>Before image cluster age</b>	<p>The number of seconds before OpenEdge reuses a before-image cluster.</p> <p>The default value is 60.</p>
<b>After image process</b>	<p>Start an After-imaging process automatically when the database starts.</p> <p>The check box is cleared by default.</p>
<b>After image stall</b>	<p>Suspend database activity if all the after-image (AI) files are filled.</p> <p>By default, if all the AI files are filled and the database cannot switch to an empty one, the database shuts down. With AI stall, you can suspend database activity, back up the full after-image file, and mark it as empty.</p> <p>The check box is cleared by default.</p>
<b>After image buffers</b>	<p>The number of after-image buffers to use.</p> <p>This parameter is useful only when running the after-image writer (AIW) process because the AIW writes the filled after-image buffers to disk, making the buffers available to other client and server processes.</p> <p>The default value is 5.</p>

**Table 2–5: Advanced database configuration properties***(1 of 3)*

Property	Description
<b>Semaphore sets</b>	<p>The number of semaphore sets available to the database broker.</p> <p>On single-processor systems, OpenEdge uses semaphores to synchronize the activities of server and self-service client processes that are connected to a database. By default, each database has an array of semaphores, one for each user or server. Each process uses its semaphore when it must wait for a shared resource. Semaphores are not used for single-user sessions or for client sessions connecting to a remote database on a server system.</p> <p>By default, this value is set to 1.</p> <p>When more than 1,000 users connect to a single database, there might be high contention for the semaphore set. If there is a lot of semaphore contention on a system, using multiple semaphore sets helps alleviate this contention and improves performance on high user counts.</p>
<b>Shared memory overflow size</b>	<p>In kilobytes, the size of the shared-memory overflow area.</p> <p>This parameter replaces the default value of the shared-memory overflow area; it does not increase it. The overflow area is appended to the shared-memory area.</p> <p>The default value is 0.</p>

**Table 2–5: Advanced database configuration properties***(2 of 3)*

Property	Description
<b>Spin lock retries</b>	<p>The number of times a process tries to acquire a latch before pausing.</p> <p>If the process cannot acquire the resource's latch, it continues the attempt. This iterative process is called spinning.</p> <p>By default, this value is set to 0.</p> <p>If the value of this property is greater than zero, a spin lock algorithm is used for shared-memory data structures. When a process has to lock a shared-memory structure, the process attempts to acquire the latch for that structure up to the specified number of times.</p> <p>If the process has not acquired the latch in the specified number of attempts, then the process pauses, or naps. The length of the pause increases gradually if the process repeatedly fails to acquire a latch. After the allotted nap time, the process wakes up and attempts to acquire the latch again. If it fails to acquire the latch, it retries up to the number of attempts specified.</p>
<b>Page writer queue delay</b>	<p>The number of milliseconds between scans of the page writer queue.</p> <p>The default value is 100.</p>
<b>Page writer queue minimum</b>	<p>The minimum number of buffers required on the page writer queue before an APW writes them to disk.</p> <p>The default value is 1.</p>
<b>Page writer scan</b>	<p>The number of buffers that an APW should scan during a buffer scan cycle.</p> <p>The default value is 1.</p>
<b>Page writer scan delay</b>	<p>The number of seconds of delay between each scan of the database buffers by the APW.</p> <p>The default value is 1.</p>
<b>Page writer max buffers</b>	<p>The maximum number of modified database buffers to write to disk during a buffer scan cycle.</p> <p>The default value is 25.</p>

**Table 2–5: Advanced database configuration properties**

(3 of 3)

Property	Description
<b>Nap interval</b>	The minimum nap time in the spin lock algorithm. The default value is 1.
<b>Nap max interval</b>	The maximum nap time in the spin lock algorithm before the nap time resets to the minimum. The default value is 1.

**Table 2–6: Agent database configuration property**

Property	Description
<b>Monitored</b>	Starts the database monitoring agent automatically. This option is selected by default.

## Editing the database default configuration

OpenEdge Management and OpenEdge Explorer provide a default database configuration. You can modify the existing configuration and then either save it as a replacement for the default configuration or create a new configuration.

---

**Note:** When you create a new configuration, however, the configuration automatically uses the pre-defined OpenEdge defaults.

---



### To edit a database default configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and Control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click the **configuration.<dbname>.defaultconfiguration**. The **Database Configuration** page appears, with the default configuration properties organized into the five categories described in [Table 2–2](#) through [Table 2–6](#).
5. Click **Edit** and make the modifications.
6. Click **Save**.

## Creating a database configuration

OpenEdge Management and OpenEdge Explorer provide a predefined default database configuration; however, you can create a new configuration by modifying the default settings.



### To create a database configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click **configuration.<dbname>.defaultconfiguration**. The **Database Configuration** page appears and displays the default configuration properties.
5. Click **Create**.
6. Type a new configuration group name in the field provided, and click **Save**. The **Database Configuration** page appears.
7. Click **Edit** to set the properties of the new database configuration. (Refer to [Table 2-2](#) through [Table 2-6](#) for a description of each property.)
8. When you finish, click **Save**.

To make this new configuration the default, you must select it as the default on the **Database Configuration** page.

To assign the new configuration to a database, see the “[Assigning a different default database configuration](#)” section on page 2-15.



## Assigning a different default database configuration

OpenEdge Management and OpenEdge Explorer provide a default database configuration. You can modify that default configuration, or you can create one or more additional configurations and assign one of them as the default.



### To assign a different default database configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for that database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click **Edit**.
5. From the **Default configuration** dropdown, select the default configuration you want to use.
6. Click **Save**.

## Viewing the server group

The server group specifies the logical collection of server processes to start.

You can apply properties to a database server group by either using the default configuration or creating and then specifying a configuration of your own choosing.

You can set or edit any of the database server group properties from OpenEdge Management or OpenEdge Explorer. The properties are organized into the **General** and **Advanced** categories.



### To view the server group:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click `servergroup.<databasename>.defaultconfiguration.defaultservergroup`. The **Database Configuration** page for the server group appears.

From the **Database Configuration** page, you can do the following:

- View a read-only display of the **General** and **Advanced** database server group properties as described in [Table 2–7](#) and [Table 2–8](#), respectively.
- Click **Edit** to modify the database server group properties. For details, see the “[Editing a database server group](#)” section on page 2–19.
- Click **Create** to create or modify a server group. For details, see the “[Creating a database server group](#)” section on page 2–19.
- Click **Delete** to delete a server group, provided it is not running. For details, see the “[Deleting a database server group](#)” section on page 2–26.

**Table 2–7: General database server group properties**

(1 of 2)

Property	Description
<b>Service name or port</b>	<p>The service name or port number through which clients may connect to the database.</p> <p>The default value is 0.</p> <p><b>Notes:</b> The service name or port number must represent a non-zero value in order to start a networked database broker.</p> <p>When you define multiple server groups for the same configuration, you must assign a unique service name or port number to each server group.</p>
<b>Number of servers</b>	<p>The maximum number of server processes that can be started to service clients.</p> <p>The default value is 0.</p>

**Table 2–7: General database server group properties**

(2 of 2)

Property	Description
<b>Message buffer size</b>	<p>The standard message buffer size in bytes.</p> <p>OpenEdge uses message buffers to move records (messages) between servers and remote clients. Records (plus 40-byte headers) larger than the message buffer size are fragmented into multiple messages. If your database records are large, increase this parameter to avoid record fragmentation.</p> <p>However, if the network works more efficiently with small messages, reduce the <b>Message Buffer size</b> value and fragment larger records.</p> <p>The default value is set dynamically based on your platform (UNIX or Windows) and the type of database (Personal, Workgroup, or Enterprise).</p>
<b>Reporting interval</b>	<p>The interval, in minutes, at which the server polls for license usage statistics that are reported in the usage report file.</p> <p>The default value is 1.</p>
<b>TCP/IP Version</b>	<p>The TCP/IP version—IPv4 or IPv6—that the database broker uses to open the database port.</p> <p>The default value is IPv4.</p>
<b>Client Type</b>	<p>One of the following client types:</p> <ul style="list-style-type: none"> <li>• <b>4GL only</b> — The server group can service requests only from ABL clients. The settings are the same as those for the <b>Both 4GL and SQL</b> option.</li> <li>• <b>SQL only</b> — The server group can service requests only from SQL clients. The settings are the same as those for the <b>Both 4GL and SQL</b> option, except that the <b>Message buffer size</b> and <b>Reporting interval</b> settings are unavailable.</li> <li>• <b>Both 4GL and SQL</b> — The server group can service requests from both ABL and SQL clients.</li> </ul> <p><b>Note:</b> Remember that the client type choice you make here is honored only when you have installed the proper licenses for that client. For example, if you selected the <b>4GL only</b> option during the OpenEdge installation, choosing the <b>SQL only</b> option or the <b>Both 4GL and SQL</b> option in the management console has no effect. However, if you chose the <b>Both 4GL and SQL</b> option during the OpenEdge installation, choosing <b>SQL only</b> in the management console is valid.</p>

**Table 2–8:      Advanced database server group properties**

<b>Property</b>	<b>Description</b>
<b>Maximum clients per server</b>	The maximum number of remote users per database server. The default value is 0.
<b>Minimum clients per server</b>	The minimum number of remote users on a server before the broker starts another server (up to the maximum number of servers). The default value is 1.
<b>Maximum dynamic port</b>	The port number that is the highest in a specified range of server port numbers accessible to a client.  You specify the lowest port number with the <b>Minimum Dynamic Port</b> property.  The default value is set dynamically based on your platform (UNIX or Windows) and the type of database (Personal, Workgroup, or Enterprise).
<b>Minimum dynamic port</b>	The port number that is the lowest in a specified range of server port numbers accessible to a client.  You specify the higher port number with the <b>Maximum dynamic port</b> property. Ports below 1025 are usually reserved for system services.  The default value is set dynamically based on your platform (Unix or Windows) and the type of database (Personal, Workgroup, or Enterprise).
<b>Host</b>	The database's host machine.

## Editing a database server group

For a database instance you added to OpenEdge Management or OpenEdge Explorer, you can edit any of the database server group properties.



### To edit a database server group:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click the `servergroup.<databasename>.defaultconfiguration.defaultservergroup` you want to edit. The **Database Configuration** page for the server group appears.
5. Click **Edit** and make the modifications. (For details about the properties in the **General** and **Advanced** categories, refer to [Table 2-7](#) and [Table 2-8](#), respectively.)
6. Click **Save**.

## Creating a database server group

You can create a new database server group from OpenEdge Management or OpenEdge Explorer.



### To create a database server group:

1. Click **Resources** in the OpenEdge Explorer console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click the `servergroup.<databasename>.defaultconfiguration.defaultservergroup` you want to edit. The **Database Configuration** page for the server group appears.
5. Click **Create**.
6. Type a new server group name in the field provided, and click **Save**. The **Database Configuration** page appears.

You can click **Edit** to set the properties of the new database server group. (For details about the properties in each category, refer to [Table 2-7](#) and [Table 2-8](#).) When you finish, click **Save**.

## Starting or stopping the database

OpenEdge Management and OpenEdge Explorer support starting and stopping both local databases and databases running under a remote AdminServer on another machine. A local database resides on the same machine as OpenEdge Management or OpenEdge Explorer.

Before you can start or stop a database, you must first connect to the AdminServer that contains a definition of the database.



### To start or stop a database:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Control**. The **Database Control** page appears.
4. Start or stop the database as follows:
  - To start the database and the agent, click **Start Database**. The status is updated to **Running** on the database's **Details** page.
  - To stop a running database, click **Stop Database**. Click **OK** to confirm that you want to stop the database. The status is updated to **Not Running** on the database's **Details** page. To restart the database, click **Start Database**. (This is the same button—the text changes to display the status opposite the one currently set.)

The database agent works only if the database is running. Therefore, when you stop a database, the agent status automatically changes to **Not Running**. However, when you start a local database whose Agent **Monitored** property is disabled, the agent does not automatically start. You must specifically click **Start Agent**.

Note that:

- You can run only one configuration for a database at a time.
- You can set the database configuration to start automatically when the AdminServer starts. For details, see the [“Starting a database automatically”](#) section on page 2–21.

## Starting a database automatically

You can set a database to start automatically when the AdminServer starts.



### To start a database automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
4. Click **Edit**.
5. Select the **Autostart** option by clicking in the checkbox.
6. Click **Save**.

## Using the background writers and watchdog process

On shared-memory systems, setting database background writer and watchdog processes can improve database performance. Background writers continuously perform overhead functions in the background.

### Background writers

There are three types of background writers:

- **Asynchronous Page Writer (APW)** — The APW improves performance by performing overhead operations in the background. These operations provide available buffers, reduce the number of buffers that OpenEdge reads before writing to disk, and reduce the overhead associated with before-image checkpoints (the process of synchronizing the buffer pool of modified blocks to the database). You can start and stop an APW at any time without shutting down the database.
- **Before-image Writer (BIW)** — The BIW improves performance by continually writing before-image buffers to disk. These writes occur in the background. BIWs are optional; however, they are highly recommended for improving I/O performance. You can run only one BIW per database. You can start and stop a BIW at any time without shutting down the database.
- **After-image Writer (AIW)** — The AIW improves performance by continually writing after-image (AI) buffers to disk soon after OpenEdge fills the buffers. The after-imaging feature lets you recover a database that was damaged when a failure caused the loss of the database or primary recovery (BI) area. When you enable after-imaging, OpenEdge writes notes to the after-image files that contain a description of all changes to the database. You can run only one AIW process per database. You can start and stop an AIW at any time without shutting down the database.

### The watchdog process

The watchdog process improves performance by cleaning up after improperly terminated processes. It releases locks, backs out any live transactions, and releases shared-memory locks. It also disconnects and cleans up the server's remote clients.



## Viewing database status

You can view the status of a database's configurations, server groups, and auxiliary processes through OpenEdge Management or OpenEdge Explorer. You can also view the status of the database monitoring agent.

---

**Note:** Most of the broker instances provide a status view, enabling you to monitor the status of multiple brokers simultaneously. You can monitor instances located on the same or different servers.

---



### To view the status of a database:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. Click **Control**. The **Database Control** page appears.
4. Review the following status information for the database:
  - **Database information** — The database name, the active port or ports, and the configuration being used by the database.
  - **Database status** — The status for the database and monitoring agent.
  - **Database auxiliary status** — The status of the After image page writer, the Before image page writer, the Auxiliary page writers, and the Watchdog process for this server group.
  - **Properties** — The status of the **Monitor** and **Enabled** settings: enabled (checked) or disabled.

## Viewing database process details

You can access real-time details and statistics that provide you with snapshot information about a database at the point you access this information from the **Control** page. Review this information to help you assess the database's performance.



### To access processing details:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. Click **Control**. The **Database Control** page appears.

4. Click a **PID** link. The following sections appear:
  - The **Process summary** section identifies the **Process name** and **Process start time**. **User id** and **Group id** values appear when UNIX-based data is shown. The **Parent pid** identifies the identifier number associated with the process that spawned this current process.
  - The **Process statistics** section presents details about the database'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 2–9 identifies and describes the fields of information presented in the **Process statistics** section.

**Table 2–9: Process statistics data**

Field	Description
<b>Resident size</b>	The physical size of the process as defined by the host system.
<b>Virtual size</b>	The virtual size of the process as defined by the host system.
<b>CPU</b>	The percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll.
<b>User time</b>	The amount of CPU time spent in the user mode since the last scheduled poll.
<b>Kernel time</b>	The amount of CPU time spent in the kernel mode since the last scheduled poll.
<b>Process time</b>	The sum of the values that appear in the <b>User time</b> and <b>Kernel time</b> fields.
<b>Weighted CPU</b>	<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 appears only when there is more than one CPU process on the system where the process is running.</p>

## Deleting a database, database configuration, or database server group

You can delete a database, provided it is not running.

You can delete a database configuration or database server group provided it is not currently in use by a database.

### Deleting a database

When you no longer need a particular database, you can delete the database and all associated configuration information from the management console. When you delete a database from the console, you are not deleting the actual, physical database files; you are only deleting the database definition from the `conmgr.properties` file. After you delete the database, it no longer appears in the console.

You cannot delete a database that is running. You must first stop the database and then delete it.



#### To delete a database from the management console:

1. Click **Resources** in the console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. Click **Delete** and then click **OK** to confirm the deletion.

### Deleting a database configuration

You can delete a database configuration from the management console.

If a configuration is running, you cannot delete it. You must first stop the configuration, and then delete it.



#### To delete a database configuration from the management console:

1. Click **Resources** in the console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. Click **Configuration**. The **Database Configuration** page appears.
4. In the **Configuration and Server Group Links** section of the page, click the database configuration you want to delete. The configuration information appears.
5. Click **Delete**, and then click **OK** to confirm the deletion.

## Deleting a database server group

You can delete a database server group from the management console.

If the configuration associated with the server group is in use, you cannot delete it. You must first stop the configuration, and then delete the server group.



### To delete a database server group:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. Click **Configuration**. The **Database Configuration** page appears.
4. In the **Configuration and Server Group Links** section of the page, click the server group you want to delete. The server group information appears.
5. Click **Delete**, and then click **OK** to confirm the deletion.

## Configuring an OpenEdge Replication-enabled database

To configure your source database and target database for OpenEdge Replication, you must configure the replication property files for both the OpenEdge Replication source database and the OpenEdge Replication target database.

In the *OpenEdge-install-directory/properties* folder, OpenEdge Replication provides two sample property files: *source.repl.properties* and *target.repl.properties*. You can save these files and modify them, using either OpenEdge Management or OpenEdge Explorer, or a text editor, so that you can use them with your source and target OpenEdge Replication databases. Because the files are samples, some of the properties and values are generic; it is up to you to customize the properties to reflect the settings you want to use with your own source and target databases.

When you use OpenEdge Management or OpenEdge Explorer to make configuration changes to the Replication server or Replication target properties, the changes are reflected in the applicable property file.

### Rules for setting properties for a Replication-enabled database

From the management console, you can configure various properties for a database enabled for Replication. The options you have in the console for configuring these Replication-related properties depend upon the following rules:

- For a database **not enabled** for Replication:
    - If the Replication property file **does not** exist, no Replication links are available.
    - If the Replication property file **does** exist but is empty, a **Server (create new)** link and an **Agent (create new)** link appear.
- 
- Note:** You can configure transition only when one of these groups exists.
- 
- If the Replication property file **does** exist and **is not** empty, then all groups defined in the property file appear.
- For a database **enabled** for Replication:
    - If the Replication property file **does not** exist, Replication-related links still appear. However, you can create only the group for which the database is enabled. In other words, you can create the Server group for source-enabled databases and the Agent group for target-enabled databases.
    - If the Replication property file **does** exist, only the groups that are applicable to the database enablement appear. For source enablement, **Server**, **ctrl-agents**, and **transition** appear. For target enablement, **Agent** and **transition** appear.
    - If the Transition group database-role is set to **reverse**, you either see the existing group for the reverse role or are allowed to create a new group for the reverse role.

## Setting Replication-related properties

If you are familiar with OpenEdge Replication and its configuration, see the following topics for details about the properties you can configure through the management console:

- [Setting OpenEdge Replication Agent properties](#)
- [Setting OpenEdge Replication Control Agent properties](#)
- [Setting OpenEdge Replication Server properties](#)
- [Setting OpenEdge Replication Transition properties](#)

If you are not familiar with OpenEdge Replication and its configuration, see the *OpenEdge Replication User Guide* for details about setting the properties. You can find the guide in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site (<http://www.psdn.com/library/kbcategory.jspa?categoryID=129>).

## Setting OpenEdge Replication Agent properties

You set OpenEdge Replication agent properties to define the configuration for the local agent running for the target database.

---

**Note:** If you are not familiar with OpenEdge Replication and its configuration, see the *OpenEdge Replication User Guide* in the Product Documentation section of the Progress Software Developer's Network Web site **before** setting the properties.

---



### To set the OpenEdge Replication Agent properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page opens.
4. In the **OpenEdge Replication Group Links** section, click **Agent**. The **Database Replication Configuration** page opens.
5. Set the following properties:
  - **Agent name** — Specifies the OpenEdge Replication agent name. By convention, the agent-name should match the name specified in the source database properties file.  
  
ALL is not allowed as an agent name. Each agent must have a unique name.
  - **Database** — Specifies the source database name.

- **Connection Timeout** — Specifies how many seconds the OpenEdge Replication agent will wait for connection from the OpenEdge Replication server before the Replication agent shuts itself down.

The seconds value must be greater than or equal to 120, and less than or equal to 86,400.

Using this property means you do not have to do a forced shutdown on your target database. If the OpenEdge Replication agent does not receive a connection attempt from the OpenEdge Replication server before the number of seconds specified have elapsed, the OpenEdge Replication agent will terminate and allow some limited system-level target database connections.

- **Polling delay minimum** — Specifies the minimum value, in milliseconds, for a polling delay.

By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

The milliseconds value must be greater than or equal to one, and less than or equal to ten.

- **Polling delay maximum** — Specifies the maximum value, in milliseconds, for a polling delay.

By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

The milliseconds value must be greater than or equal to 500 and less than or equal to 1000.

- **Agent minimum port** — Specifies the minimum TCP port number. The agent selects a port in a range between the values specified by the minimum port and the maximum port.

The minimum port number value must be greater than 1024 and less than the value of the maximum port.

- **Agent maximum port** — Specifies the maximum TCP port number for the agent.

The maximum port number value must be greater than the minimum port + 1, and less than the maximum allowable port number on the system.

For UNIX the maximum port number is 65534. For Windows the maximum port number is 32765.

- **Other arguments** — Any other arguments for which there is no defined property available.

6. Click **Save**.

## Setting OpenEdge Replication Server properties

You set the OpenEdge Replication Server properties to specify server and agent property values.

---

**Note:** If you are not familiar with OpenEdge Replication and its configuration, see the *OpenEdge Replication User Guide* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site **before** setting the properties.

---

The server properties are divided into the following categories:

- General
- Advanced



### To set the OpenEdge Replication Server properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page opens.
4. In the **OpenEdge Replication Group Links** section, click **Server**. The **Database Replication Configuration** page opens.
5. Set the following **General** properties:

- **Database** — Specifies the source database name.
- **Defer Agent Startup** — Specifies for how long, in minutes, the server attempts to connect to an agent if the first connection attempt is unsuccessful. The minutes value must be greater than or equal to zero and less than or equal to 10080.
- **Keep Alive** — Specifies a timeout period for communications between a server and its agents. If a connection between the server and the agent is not verified before the timeout expires, failure recovery begins.

By default, this property is enabled and has a value of 300 seconds. The minimum value is 60 seconds; there is no maximum value.

- **Server Transition Timeout** — Specifies the number of seconds the target database waits before it performs auto-transition. This property is ignored when transition=manual.

The value is incremented by the sum of the connect-timeout for all configured agents.



- **Server Transition** — Specifies how to transition the target database to a normal database.

If a synchronous agent or an asynchronous critical agent cannot reconnect with the source database within the time specified in the transition-timeout property, the following will occur:

- **auto** — The agent automatically transitions the target database.
- **manual** — The agent listens for the server until it reconnects with the source database or the DSRUTIL command is executed. The DSRUTIL db-name -C transition command transitions the target database.

6. Set the following **Advanced** properties:

- **Other arguments** — Any other arguments for which there is no defined property available.
- **Schema Lock Action** — Specifies the action an agent takes if an exclusive schema lock is not granted. Possible actions are:
  - **wait** — The agent waits until the exclusive schema lock is granted. The server blocks until the exclusive schema lock is granted.
  - **force** — The agent attempts to acquire the exclusive schema lock five times. If the fifth attempt fails, the agent disconnects all users from the target and makes another attempt. If the last attempt fails, the server and all agents terminate. When schema update activity completes, the server and target can be restarted.
- **Polling delay minimum** — Specifies the minimum value, in milliseconds, for a polling delay. By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

The value must be greater than or equal to one and less than or equal to ten.

- **Polling delay maximum** — Specifies the maximum value, in milliseconds, for a polling delay. By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

The value must be greater than 500 and less than 1000.

- **Server port minimum** — The minimum port available.
- **Server port maximum** — The maximum port available.
- **Agent Shutdown Action** — Specifies the action an agent takes during a shutdown when the Replication server ends. Possible actions are:
  - **recovery** — The agent remains active but in a standby state waiting for the replication server to reconnect.
  - **normal** — The agent terminates; the target database stays up.

Then click **Save**.

## Setting OpenEdge Replication Control Agent properties

You set the OpenEdge Replication control agent properties to define for the server which agents it will contact, where it will contact them, and how the agent should perform. Each server group supports up to two control agents.

---

**Note:** If you are not familiar with OpenEdge Replication and its configuration, see the *OpenEdge Replication User Guide* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site **before** setting the properties.

---



### To set the OpenEdge Replication Control Agent properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page opens.
4. In the **OpenEdge Replication Group Links** section, click **Control-Agent**. The **Database Replication Configuration** page opens.
5. Click **Edit**.
6. Set the properties as follows:
  - **Database** — Specifies the source database name.
  - **Host** — Specifies to the server which host the agent will start on.
  - **Port** — Specifies which port the server uses to connect to the target database. The port number specified must be the same port specified with the `-S` parameter when the target database broker was started.
  - **Connection Timeout** — Specifies how many seconds the OpenEdge Replication agent will wait for a connection from the OpenEdge Replication server before the Replication Agent shuts itself down.

The seconds value must be greater than or equal to 120, and less than or equal to 86,400.

Using this property means you do not have to do a forced shutdown on your target database. If the OpenEdge Replication agent does not receive a connection attempt from the OpenEdge Replication server before the number of seconds specified elapses, the OpenEdge Replication agent will terminate and allow some limited system-level target database connections.

This property is also used by the server when reconnecting to the agent after communication has been lost.

- **Synchronous or Asynchronous** — Specifies the type of replication, either asynchronous (**async**) or synchronous (**sync**).

- **Maximum message** — The maximum number of bytes used for the TCP/IP communication messages.
- **Replication method** — Asynchronous or synchronous.
- **Critical agent** — Specifies whether the agent is critical. (Select the checkbox for a critical agent.)

A critical agent is an asynchronous agent for the target database that can become the source database if the source database becomes unavailable.

- **Other arguments** — Any other arguments for which there is no defined property available.

7. Click **Save**.

## Setting OpenEdge Replication Transition properties

You set the Transition properties to specify the transition process values, which are used when the target database becomes a normal database after a failure.

---

**Note:** If you are not familiar with OpenEdge Replication and its configuration, see the *OpenEdge Replication User Guide* in the Product Documentation section of the Progress Software Developer's Network Web site **before** setting the properties.

---

The transition properties are divided into the following categories:

- General
- Start Up
- After Image
- Back Up



### To set the OpenEdge Replication Transition properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for the database appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page opens.
4. In the **OpenEdge Replication Group Links** section, click **Transition**. The **Database Replication Configuration** page opens.

5. Set the following **General** properties:

- **Database Role:**

- **Reverse** — The role of the database is reversed: A source database becomes a target database, and a target database becomes a source database.
- **Normal** — The role of the database becomes that of a normal database; the database is no longer enabled for replication once the transition is performed. This is the default value.

- **Responsibility:**

- **Primary** — This is the primary database.
- **Secondary** — This is the secondary database.

- **Transition to Agents** — Transitions to the first agent in the list when a failure occurs. If the first agent is not available, transition to the second agent in the list.

A valid value for this property is any configured agent name; separate the names by a comma if you are listing more than one. For example, `agent1,agent2`.

6. Set the following **Start Up** properties:

- **Restart after transition** — The database can be automatically restarted after transition is performed. When you select this option, the `*-startup-arguments` properties must be supplied, or the database startup will fail:
  - If the database role is normal, you must specify the `normal-startup-arguments`.
  - If the database role is reversed, you must specify the `source-startup-arguments` and the `target-startup-arguments`.
- **Source startup arguments** — If the database is transitioned to a source database, these arguments are used when the database is started. The arguments are appended to the PROSERVE command (used to start the database).

In most cases, the only argument specified here should be `-pf` followed by a parameter file name; for example, `-pf db-name.source.pf`.

Because the database is a source, you must also specify the `-DBService replserv` argument, as an indication to the broker to start the Replication server.

- **Target startup arguments** — If the database is transitioned to a target database, these arguments are used when the database is started. The arguments will be appended to the PROSERVE command.

In most cases, the only argument specified here should be `-pf` followed by a parameter file name; for example, `-pf db-name.target.pf`.

Because the database is a target, you must also specify the `-DBService replagent` `-S port-number` or `service-name` arguments, as an indication to the broker to start the Replication agent and to listen on the TCP/IP port specified with `-S port-number` or `service-name`.

- **Normal startup arguments** — If the database is transitioned to a normal database, these arguments are used when the database is started. The arguments will be appended to the PROSERVE command.

In most cases, the only argument specified here should be `-pf` followed by a parameter file name; for example, `-pf db-name.normal.pf`.

7. Set the following **After image** properties:

- **Begin After imaging after transition** — Select the option to automatically begin AI after a target-to-source database transition. After-imaging can be started for a database that has AI areas.
- **Automatically add After imaging extents** — This property instructs the transition process to automatically add AI areas to the database if:
  - It is transitioned to a source database.
  - There are currently no AI areas for the database.

If you do not select the option, AI areas are not added to the database. If you do select the option, AI areas are automatically added to the database by using the structure file specified in the `ai-structure-file` property.

- **Structure file containing After image area definitions** — The name of the structure file that contains the list of AI areas to add.

8. Set the following **Back Up** properties:

- **Backup method** — Specifies the backup method performed before AI is enabled. The following values are valid:
  - **Mark** — Marks the database backed up by using the `rfutil db-name -C mark backedup` command. Marking the database as backed up does not allow future AI extents to be used when recovering from a disaster
  - **Full off-line** — Backs up the database offline by using the OpenEdge PROBKUP utility. The backup is performed in two steps. The first backup is a full backup, which is performed before AI is enabled for the database. The second backup is an incremental backup, which is performed after AI is enabled, and after the role of the database is changed.
  - **Full on-line** — This backup is performed after the database is restarted, which occurs after the database has been transitioned.

- **Full on-line and off-line backup arguments** — The arguments required for the full online and offline backups that are performed for the database. At a minimum, the target file or device must be specified in these arguments for both online and offline backups.

To avoid overwriting a backup, do not use the same target file or device for both the backup and the incremental backup.

Do not use backup validation parameters (such as `-vp` and `-vf`).

The arguments should begin with `device-name`.

- **Incremental backup arguments** — The arguments required for the offline incremental backup performed after AI is enabled and the database's role is reversed.

The arguments should begin with `device-name`.

## **Configuring DataServers**

---

You can configure DataServer property settings and add new DataServer instances in OpenEdge Management and OpenEdge Explorer. You can also start, stop, and delete DataServer instances, as well as view their status and log files.

This chapter provides the following information:

- [ODBC DataServer configuration and administration](#)
- [Oracle DataServer configuration and administration](#)
- [MS SQL Server DataServer configuration and administration](#)

## ODBC DataServer configuration and administration

The OpenEdge® DataServer for ODBC allows the OpenEdge Application Development Environment (ADE) and applications created with OpenEdge in Windows to access the ODBC-compliant databases DB2 and Sybase.

You can use OpenEdge Management or OpenEdge Explorer to configure and administer the configuration of an OpenEdge DataServer for ODBC. The DataServer for ODBC installation provides one predefined DataServer broker (odbbroker1) and one predefined NameServer (NS1). You can use these predefined components as a starting point for creating and configuring additional DataServer brokers, and, if needed, NameServers. Each broker is referred to as an instance.

### Working with the ODBC DataServer

To work with the ODBC DataServer, double-click the **ODBC DataServer** folder from the management console list frame. There is one pre-defined default ODBC DataServer broker (osbroker1).

You can perform the following ODBC DataServer actions:

- [Configuring an ODBC DataServer broker](#)
- [Editing an ODBC DataServer broker configuration](#)
- [Creating an ODBC DataServer broker](#)
- [Starting an ODBC DataServer broker](#)
- [Starting an ODBC DataServer broker automatically](#)
- [Stopping an ODBC DataServer broker](#)
- [Viewing the status of an ODBC DataServer broker](#)
- [Deleting an ODBC DataServer broker](#)
- [Viewing the status of an ODBC DataServer broker](#)



## Configuring an ODBC DataServer broker

You can view or modify the following default configuration properties of an ODBC DataServer broker from the management console:

- Broker
- Server
- Environment variables

You can also create a new ODBC DataServer broker, view a broker's status, delete a broker you no longer need, or view the broker's log files.

Each of these tasks is described in the following sections.

### Viewing or modifying ODBC DataServer broker properties

You can view or modify ODBC DataServer Broker properties in the following categories: general, owner information, controlling NameServer, Data Service, logging setting, and advanced features



#### To view or modify broker properties for an ODBC DataServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **ODBC DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the ODBC broker properties as described in [Table 3–1](#).
- Click **Edit** to modify the broker properties. For details, see the [“Editing an ODBC DataServer broker configuration”](#) section on page 3–8.

**Table 3–1: ODBC DataServer broker properties**

(1 of 4)

Property	Description
<b>General</b>	
<b>Working directory</b>	Your general working directory. The default value is @WorkPath.
<b>Operating mode</b>	For DataServer brokers, the operating mode is state-aware. The default value is state-aware.
<b>Port number</b>	The type of the TCP/IP listening port that the DataServer broker uses. The default value is 3050.

**Table 3–1: ODBC DataServer broker properties***(2 of 4)*

Property	Description
<b>Auto start</b>	Starts the DataServer automatically when the AdminServer starts. The Auto start check box is cleared by default.
<b>Owner Information</b>	
<b>Group name</b>	N/A
<b>Username</b>	Your user name. The default is blank.
<b>Password</b>	Your password. The default is blank.
<b>Controlling NameServer</b>	
<b>Register with NameServer</b>	A checkbox that indicates whether to register the DataServer with a controlling NameServer.
<b>Controlling NameServer</b>	Identifies the NameServer that the DataServer uses to find a broker that supports the correct application services. (If you did not select to register the DataServer with a NameServer, the field is dimmed and unavailable.)
<b>Registration mode</b>	<p>How the broker specifies its hostname when registering with its controlling NameServer. This hostname information is passed onto a client application when it attempts to connect to a Data Service that the broker supports. The choices are:</p> <ul style="list-style-type: none"><li>• <b>Register IP</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li><li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li><li>• <b>Register-HostName</b> — The broker registers with the values specified in the hostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li></ul>
<b>Registration host name</b>	Identifies the hostname value to be used if the <b>Registration mode</b> property is set to <b>Register-HostName</b> .

**Table 3–1: ODBC DataServer broker properties**

(3 of 4)

Property	Description
<b>Data Service</b>	
<b>Data service name list</b>	Data services are services that are provided by a selected broker's servers. The default service name is the broker name. The default value is blank.
<b>Supports default service</b>	A check box that indicates if the ODBC broker supports the default service. Select the check box to have the broker support the default service, or click to clear the check box and cancel support for the default service. The <b>Supports default service</b> check box is cleared by default.
<b>Logging Setting</b>	
<b>Broker log filename</b>	The general broker log filename. The default is @{\WorkPath}\oddbroker1.broker.log.
<b>Broker logging level</b>	<p>The amount of information written to the broker log. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — No log file is written.</li> <li>• <b>Error Only</b> — Only error-related information is written to the log file, which reduces the size of the log file.</li> <li>• <b>Basic</b> — Only some information is written to the log file.</li> <li>• <b>Verbose</b> — All broker activity is written to the log file. This might produce large files that use up disk space.</li> <li>• <b>Extended</b> — Information in addition to <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	Deselect to create a new broker log file each time a broker is started, even if the broker log file specified already exists. The <b>Append to broker log file</b> check box is selected by default.
<b>Broker logging entry types</b>	A single entry or comma delimited list of logging entry types.
<b>Broker log file threshold size</b>	Specifies a limit on how big the log file can get (in bytes). When the log is full, the broker creates a new log file with a sequence number.
<b>Maximum number of broker log files</b>	Limit on how many broker log files will be kept on the system when there is a threshold limit. It represents the number of log files including the current on and includes any rolled over log files that already exist when the process started.

Table 3–1: ODBC DataServer broker properties

(4 of 4)

Property	Description
<b>Advanced Features</b>	
<b>Maximum client instances</b>	<p>The maximum number of client connections that the broker can support concurrently.</p> <p>The default value of 512 that is set for this parameter is high enough to ensure that the number of client connections is unlimited. Realistically, however, the system-level resources needed to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you might need to reduce the maximum number of client connections to a more reasonable value. Note, however, that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring (and starting) multiple brokers to handle higher client loads alleviates the problem.</p>
<b>Priority weight (0-100)</b>	<p>An integer value between 0 and 100 that influences the share of the workload that the selected broker receives. The larger the value, the heavier the load that is distributed to the broker. Using the <b>Priority weight</b> value, the NameServer distributes client requests to the same DataServer service across all DataServer brokers. The default value is 0.</p>
<b>Registration retry (in seconds)</b>	<p>After registering with its controlling NameServer, the DataServer broker periodically sends "keep-alive" messages to the NameServer to let the NameServer know that the broker is still active. The <b>Registration retry</b> value is the number of seconds that pass between "keep-alive" messages. The default value is 30.</p>

## Viewing or modifying ODBC DataServer broker server properties

You can view or modify ODBC DataServer broker Server properties in the following categories: general and logging setting.



### To view or modify server properties for an ODBC broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and Control** section of the page, click **Configuration**. The **ODBC DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the ODBC server properties as described in [Table 3–2](#).
- Click **Edit** to modify the default server properties. For details, see the “[Starting an Oracle DataServer broker](#)” section on page 3–21.

**Table 3–2: ODBC DataServer server properties**

Property	Description
<b>General</b>	
<b>Server executable file</b>	The pathname of the DataServer executable that the broker runs. The default for the DataServer for ODBC in Windows is:  @{Startup\DLCL}bin\_odbsrv.exe
<b>Server startup parameters</b>	The startup parameters for the server. You should not modify the default parameters, which are:  -svub -S -X -N TCP -U X -P x -hs 0 -s 40
<b>Minimum port number</b>	The minimum port number that you want to make available for each DataServer broker. The default is blank.
<b>Maximum port number</b>	The maximum port number that you want to make available for each DataServer broker. The default is blank.
<b>Logging Setting</b>	
<b>Server log filename</b>	The DataServer log filename. Enter a valid path name.  Installed Default: @{WorkPath}\odbbroker1.server.log

## Setting ODBC DataServer broker environment variables

You can set environment variables for the DataServer broker for ODBC. You can set one or more environment variables that:

- Are accessible from your ABL application
- Affect the environment in which your application runs

For the ODBC DataServer, there are no required environment variables in Windows.



### To set ODBC DataServer broker environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker whose environment variables you want to set. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The **Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.
7. Click **Save** when you finish adding the variables and their values.

## Editing an ODBC DataServer broker configuration

You can edit an ODBC DataServer broker configuration from the management console.



### To edit an ODBC DataServer broker configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker whose configuration you want to edit. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**. (For details about the broker and server properties, see [Table 3-1](#) and [Table 3-2](#), respectively.)
5. Make the configuration changes, and then click **Save**.

## Creating an ODBC DataServer broker

OpenEdge Management and OpenEdge Explorer provide a predefined default ODBC DataServer configuration; however, you can create a new configuration by modifying the default settings.



### To create an ODBC DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **ODBC DataServer**. The **ODBC DataServer Configuration** page appears.
4. Type the name of the new ODBC DataServer broker in the field provided.
5. Click **Save**. The **ODBC DataServer Configuration** page appears, allowing you to establish the broker's properties.

## Deleting an ODBC DataServer broker

You can delete an inactive ODBC DataServer broker from the management console.

If a broker is running, you cannot delete it. (The **Delete** button is not available.) You must first stop the broker and then delete it.



### To delete an ODBC DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker you want to delete. The **Details** page for the broker appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Starting an ODBC DataServer broker

You can start an ODBC DataServer broker from the management console, or you can choose to start the broker automatically when the AdminServer starts. You can also stop the broker.



### To start an ODBC DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker you want to start. The **Details** page for the broker appears in the detail frame.

3. Click **Control**. The broker's **Control** page appears.
4. Click **Start ODBC DataServer**.

## Starting an ODBC DataServer broker automatically

You can set an ODBC DataServer broker to start automatically when the AdminServer starts.



### To start an ODBC DataServer broker automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker you want to start automatically. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The **ODBC DataServer Configuration** page appears.
4. Click **Edit**.
5. Select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.

## Stopping an ODBC DataServer broker

You can stop a ODBC DataServer broker from the management console.



### To stop an ODBC DataServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the ODBC DataServer broker you want to stop. The **Details** page for the broker appears in the detail frame.
3. Click **Control**. The **ODBC DataServer Control** page appears.
4. Click **Stop ODBC DataServer**, and then click **OK** to confirm the deletion.

## Viewing the status of an ODBC DataServer broker

You can view the status of any active DataServer broker, server groups, and auxiliary processes through the management console. Most of the broker instances provide a status view, enabling you to monitor the status of multiple brokers simultaneously.

To see the status of a broker, the broker must be running or have been running.





**To view the status of an ODBC DataServer broker:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.
3. Click **Status**. A Status summary and status details appear and provide the information described in [Table 3–3](#) and [Table 3–4](#), respectively.

**Table 3–3: ODBC DataServer broker status summary**

Property	Description
<b>Host</b>	The broker's host machine.
<b>Broker Name</b>	The name of the broker whose status you are viewing.
<b>Operating Mode</b>	For DataServer brokers, the operating mode is state-aware.
<b>Broker Status</b>	The current state of the broker.
<b>Broker Port</b>	The TCP/IP port number that the broker listens to.
<b>Broker PID</b>	The process ID of the broker.
<b>Active Servers</b>	The number of active servers.
<b>Busy Servers</b>	N/A
<b>Locked Servers</b>	N/A
<b>Available Servers</b>	N/A
<b>Active Clients (now, peak)</b>	N/A
<b>Client Queue Path (cur, max)</b>	The number of clients waiting for agents to become available to service their request. The current value ( <b>cur</b> ) represents the number of waiting clients at the moment the status is displayed, and the maximum value ( <b>max</b> ) represents the largest number of clients waiting concurrently since the broker started.
<b>Total Requests</b>	N/A
<b>Rq Wait (max, avg)</b>	N/A
<b>Rq Duration (max, avg)</b>	N/A

**Table 3–4: ODBC DataServer broker status details**

Property	Description
<b>PID</b>	The process id of the server process
<b>State</b>	The current state of the server process
<b>Port</b>	The TCP/IP port number used by the server
<b>nRq</b>	N/A
<b>nRcvd</b>	N/A
<b>nSent</b>	N/A
<b>Started</b>	The time stamp indicating when the server process is started
<b>Last Change</b>	The time stamp indicating when the server process last changed state

## Viewing the ODBC DataServer log files

You can view the log files for an ODBC DataServer by using the log file viewer. The log file viewer allows you to examine the ODBC DataServer log files through an HTML interface.



**To access the log file viewer:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.
3. Click **Log File Viewer of Broker** or **Log File Viewer of Servers**.

You can use the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display the previous values.

- Click **Go To** to control which numbered entry in the log file the viewer display begins. 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 does 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, regardless of 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 **Previous** 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**.
- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.
- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## Oracle DataServer configuration and administration

The OpenEdge® DataServer for Oracle allows the OpenEdge Application Development Environment (ADE) and applications created with OpenEdge in Windows to access certain Oracle databases.

The DataServer for Oracle is also supported on several UNIX and LINUX platforms. For details, see <http://www.psdn.com/library/entry!default.jsps?categoryID=1592&externalID=3233&fromSearchPage=true>

You can use OpenEdge Management or OpenEdge Explorer to establish and administer the configuration of a Progress OpenEdge DataServer for Oracle. The DataServer for Oracle installation provides one predefined DataServer broker (orabroker1). You can use this predefined component as a starting point for creating and configuring additional DataServer brokers. Each broker is referred to as an instance.

### Working with the Oracle DataServer broker

To work with the Oracle DataServer, double-click the **Oracle DataServer** folder from the management console list frame. There is one pre-defined default Oracle DataServer (orabroker1).

You can perform the following Oracle DataServer actions:

- [Configuring an Oracle DataServer broker](#)
- [Editing an Oracle DataServer broker configuration](#)
- [Creating an Oracle DataServer broker](#)
- [Starting an Oracle DataServer broker](#)
- [Starting an Oracle DataServer broker automatically](#)
- [Stopping an Oracle DataServer broker](#)
- [Viewing the status of an Oracle DataServer broker](#)
- [Deleting an Oracle DataServer broker](#)
- [Viewing the ODBC DataServer log files](#)

## Configuring an Oracle DataServer broker

You can view or modify the following default configuration properties of an Oracle DataServer broker from OpenEdge Management or OpenEdge Explorer:

- Broker
- Server
- Environment variables

You can also create a new Oracle DataServer broker, view a broker's status, delete a broker you no longer need, or view the broker's log files.

Each of these tasks is described in the following sections.

### Viewing or modifying Oracle DataServer broker properties

You can view or modify Oracle DataServer broker properties in the following categories: general properties, owner information, controlling NameServer information, Data Service, setting logging properties, and advanced features.



#### To view or modify broker properties for an Oracle broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **Oracle DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the Oracle broker properties as described in [Table 3–5](#).
- Click **Edit** to modify the broker properties. For details, see the [“Editing an Oracle DataServer broker configuration”](#) section on page 3–20.

**Table 3–5: Oracle DataServer broker properties**

(1 of 4)

Property	Description
<b>General</b>	
<b>Working directory</b>	Your general working directory. The default value is @WorkPath.
<b>Operating mode</b>	For DataServer brokers, the operating mode is state-aware. The default value is state-aware.
<b>Port number</b>	The type of the TCP/IP listening port that the DataServer broker uses. The default value is 3050.

**Table 3–5: Oracle DataServer broker properties***(2 of 4)*

Property	Description
<b>Auto start</b>	Starts the DataServer automatically when the AdminServer starts. The <b>Auto start</b> check box is cleared by default.
<b>Owner Information</b>	
<b>Group name</b>	(UNIX only) The name of the group.
<b>Username</b>	Your user name. The default is blank.
<b>Password</b>	Your password. The default is blank.
<b>Controlling NameServer</b>	
<b>Register with NameServer</b>	A checkbox that indicates whether to register the DataServer with a controlling NameServer.
<b>Controlling NameServer</b>	Identifies the NameServer that the DataServer uses to find a broker that supports the correct application services. (If you did not select to register the DataServer with a NameServer, the field is dimmed and unavailable.)
<b>Registration mode</b>	<p>How the broker specifies its hostname when registering with its controlling NameServer. This hostname information is passed onto a client application when it attempts to connect to a Data Service that the broker supports. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>Register IP</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li> <li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li> <li>• <b>Register-HostName</b> — The broker registers with the values specified in the hostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li> </ul>
<b>Registration host name</b>	Identifies the hostname value to be used if the <b>Registration mode</b> property is set to <b>Register-HostName</b> .

**Table 3–5: Oracle DataServer broker properties***(3 of 4)*

Property	Description
<b>Data Service</b>	
<b>Data service name list</b>	Data services are services that are provided by a selected broker's servers. The default service name is the broker name. The default value is blank.
<b>Supports default service</b>	A check box that indicates whether the ODBC broker supports the default service. Select the check box to have the broker support the default service, or click to clear the check box and cancel support for the default service. The <b>Supports default service</b> check box is cleared by default.
<b>Logging Setting</b>	
<b>Broker log filename</b>	The general broker log filename. The default is @{\WorkPath}\orabroker1.broker.log.
<b>Broker logging level</b>	<p>The amount of information written to the broker log. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — No log file is written.</li> <li>• <b>Error Only</b> — Only error-related information is written to the log file, which reduces the size of the log file.</li> <li>• <b>Basic</b> — Only some information is written to the log file.</li> <li>• <b>Verbose</b> — All broker activity is written to the log file. This might produce large files that use up disk space.</li> <li>• <b>Extended</b> — Information in addition to <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	Create a new broker log file each time a broker is started, even if the broker log file specified already exists. The <b>Append to broker log</b> file check box is selected by default.
<b>Broker logging entry types</b>	A single entry or comma delimited list of logging entry types.
<b>Broker log file threshold size</b>	Specifies a limit on how big the log file can get (in bytes). When the log is full, the broker creates a new log file with a sequence number.
<b>Maximum number of broker log files</b>	Limit on how many broker log files will be kept on the system when there is a threshold limit. It represents the number of log files including the current on and includes any rolled over log files that already exist when the process started.

**Table 3–5: Oracle DataServer broker properties**

(4 of 4)

Property	Description
<b>Advanced Features</b>	
<b>Maximum client instances</b>	<p>The maximum number of client connections that the broker can support concurrently.</p> <p>The default value of 512 that is set for this parameter is high enough to ensure that the number of client connections is unlimited. Realistically, however, the system-level resources needed to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you might need to reduce the maximum number of client connections to a more reasonable value. Note, however, that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads alleviates the problem.</p>
<b>Priority weight (0-100)</b>	An integer value between 0 and 100 that influences the share of the workload that the selected broker receives. The larger the value, the heavier the load that is distributed to the broker. Using the <b>Priority weight</b> value, the NameServer distributes client requests to the same DataServer service across all DataServer brokers. The default value is 0.
<b>Registration retry (in seconds)</b>	After registering with its controlling NameServer, the DataServer broker periodically sends "keep-alive" messages to the NameServer to let the NameServer know that the broker is still active. The <b>Registration retry</b> value is the number of seconds that pass between "keep-alive" messages. The default value is 30.

### Viewing or modifying Oracle DataServer server properties

You can view or modify Oracle DataServer server properties in the following categories: General and Logging Setting.



#### To view or modify server properties for an Oracle broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.



3. In the **Command and control** section of the page, click **Configuration**. The **Oracle DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the Oracle server properties as described in [Table 3–6](#).
- Click **Edit** to modify the server properties. For details, see the “[Editing an Oracle DataServer broker configuration](#)” section on page 3–20.

**Table 3–6: Oracle DataServer server properties**

Server properties	Description
<b>General</b>	
<b>Server executable file</b>	<p>The pathname of the DataServer executable that the broker runs. The defaults for the DataServer for Oracle on the Windows and UNIX platforms are as follows:</p> <ul style="list-style-type: none"> <li>• Windows: @{Startup\DLC}\bin\_orasrv.exe</li> <li>• UNIX: <i>OpenEdge-install-dir/_orasrv</i></li> </ul>
<b>Server startup parameters</b>	<p>The startup parameters for the server. You should not modify the default parameters, which are:</p> <p><code>-svub -S -X -N TCP -U X -P x -hs 0 -s 40</code></p>
<b>Minimum port number</b>	The minimum port number that you want to make available for each DataServer broker. The default is blank.
<b>Maximum port number</b>	The maximum port number that you want to make available for each DataServer broker. The default is blank.
<b>Logging Setting</b>	
<b>Server log filename</b>	<p>The DataServer log filename. Enter a valid path name.</p> <p>Installed Default: @{WorkPath}\orabroker1.server.log</p>

### Setting Oracle DataServer broker environment variables

You can set environment variables for the DataServer broker for Oracle. You can set one or more environment variables that:

- Are accessible from your ABL application
- Affect the environment in which your application runs

For the Oracle DataServer, there are no required environment variables in Windows.

**To set Oracle DataServer broker environment variables:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker whose environment variables you want to set. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.
7. Click **Save** when you finish adding the variables and their values.

## Editing an Oracle DataServer broker configuration

You can edit an Oracle DataServer configuration from the management console.

**To edit an Oracle DataServer broker configuration:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker whose configuration you want to edit. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**. (For details about the broker or server properties, see [Table 3–5](#) and [Table 3–6](#), respectively.)
5. Make the configuration changes, and then click **Save**.

## Creating an Oracle DataServer broker

OpenEdge Management and OpenEdge Explorer provide a predefined default Oracle DataServer configuration; however, you can create a new configuration by modifying the default settings.

**To create an Oracle DataServer broker:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **Oracle DataServer**. The **Oracle DataServer Configuration** page appears.

4. Type the name of the new Oracle DataServer in the field provided.
5. Click **Save**. The **Oracle DataServer Configuration** page appears, allowing you to configure the broker's properties.

## Deleting an Oracle DataServer broker

You can delete an inactive Oracle DataServer broker from the management console.

If a broker is running, you cannot delete it. (The **Delete** button is not available.) You must first stop the broker and then delete it.



### To delete an Oracle DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker you want to delete. The **Details** page for the broker appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Starting an Oracle DataServer broker

You can start an Oracle DataServer broker from the management console.



### To start an Oracle DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker you want to start. The **Details** page for the broker appears in the detail frame.
3. Click **Control**. The broker's **Control** page appears.
4. Click **Start Oracle DataServer**.

## Starting an Oracle DataServer broker automatically

You can set a Oracle DataServer broker to start automatically when the AdminServer starts.



### To start an Oracle DataServer broker automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker you want to start. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The **Oracle DataServer Configuration** page appears.

4. Click **Edit**.
5. Select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.

## Stopping an Oracle DataServer broker

You can stop an Oracle DataServer broker from the management console.



### To stop an Oracle DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Oracle DataServer broker you want to stop. The **Details** page for the broker appears in the detail frame.
3. Click **Control**. The **Oracle DataServer Control** page appears.
4. Click **Stop Oracle DataServer**, and then click **OK** to confirm the deletion.

## Viewing the status of an Oracle DataServer broker

You can view the status of any active Oracle DataServer broker, server groups, and auxiliary processes through the management console. Most of the broker instances provide a status view, enabling you to monitor the status of multiple brokers simultaneously.

To see the status of a broker, the broker must be running or have been running.



### To view the status of an Oracle DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.
3. Click **Status**. A Status summary and summary details appear and provide the information described in [Table 3-7](#) and [Table 3-8](#).

**Table 3-7: Oracle DataServer broker property status summary** *(1 of 2)*

Property	Description
<b>Host</b>	The broker's host machine.
<b>Broker Name</b>	The name of the broker whose status you are viewing.
<b>Operating Mode</b>	For DataServer brokers, the operating mode is state-aware.

**Table 3–7: Oracle DataServer broker property status summary** (2 of 2)

Property	Description
<b>Broker Status</b>	The current state of the broker.
<b>Broker Port</b>	The TCP/IP port number that the broker listens to.
<b>Broker PID</b>	The process ID of the broker.
<b>Active Servers</b>	The number of active servers.
<b>Busy Servers</b>	N/A
<b>Locked Servers</b>	N/A
<b>Available Servers</b>	N/A
<b>Active Clients (now, peak)</b>	N/A
<b>Client Queue Path (cur, max)</b>	The number of clients waiting for agents to become available to service their request. The current value ( <b>cur</b> ) represents the number of waiting clients at the moment the status is displayed, and the maximum value ( <b>max</b> ) represents the largest number of clients waiting concurrently since the broker started.
<b>Total Requests</b>	N/A
<b>Rq Wait (max, avg)</b>	N/A
<b>Rq Duration (max, avg)</b>	N/A

**Table 3–8: Oracle DataServer broker status details**

Property	Description
<b>PID</b>	The process id of the server process
<b>State</b>	The current state of the server process
<b>Port</b>	The TCP/IP port number used by the server
<b>nRq</b>	N/A
<b>nRcvd</b>	N/A
<b>nSent</b>	N/A
<b>Started</b>	The time stamp indicating when the server process is started
<b>Last Change</b>	The time stamp indicating when the server process last changed state

## Viewing the Oracle DataServer broker log files

You can view the log files for an Oracle DataServer broker by using the log file viewer. The log file viewer allows you to examine the Oracle DataServer log files through an HTML interface.



### To access the log file viewer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.
3. Click **Log File Viewer of Broker** or **Log File Viewer of Servers**.

You can use the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display the previous values.
- Click **Go To** to control which numbered entry in the log file the viewer display begins. 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, regardless of 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 **Previous** 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**.

- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.
- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## MS SQL Server DataServer configuration and administration

The OpenEdge® DataServer for MS SQL Server allows the OpenEdge Application Development Environment (ADE) and applications created with OpenEdge in Windows to access Microsoft SQL Server.

You can use OpenEdge Explorer to configure and administer the configuration of an OpenEdge DataServer for SQL Server. The DataServer for SQL Server installation provides one predefined DataServer broker (mssbroker1). You can use this predefined component as a starting point for creating and configuring additional DataServer brokers. Each broker is referred to as an instance.

### Working with the MS SQL Server DataServer broker

To work with the MS SQL Server DataServer, double-click the **MSS DataServer** folder from the OpenEdge Explorer list frame. There is one pre-defined default MS SQL Server DataServer (mssbroker1).

You can perform the following MS SQL Server actions:

- [Configuring an MS SQL Server DataServer broker](#)
- [Editing an MS SQL Server DataServer broker configuration](#)
- [Creating an MS SQL Server DataServer broker](#)
- [Starting an MS SQL Server DataServer broker](#)
- [Starting an MS SQL Server DataServer broker automatically](#)
- [Stopping an MS SQL Server DataServer broker](#)
- [Viewing the MS SQL Server DataServer log files](#)
- [Deleting an MS SQL Server DataServer broker](#)
- [Viewing the MS SQL Server DataServer log files](#)



## Configuring an MS SQL Server DataServer broker

You can view or modify the following configuration properties of an MS SQL Server DataServer from the management console:

- Broker
- Server
- Environment variables

You can also create a new MS SQL Server DataServer broker, view a broker's status, or delete a broker you no longer need.

Each of these tasks is described in the following sections.

### Viewing or modifying MS SQL Server DataServer broker properties

You can view or modify MS SQL Server DataServer Broker properties in the following categories: general properties, owner information, controlling NameServer information, Data Service, setting logging properties, and advanced features.



#### To view or modify broker properties for an MS SQL Server broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and Control** section of the page, click **Configuration**. The **MSS DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the MS SQL Server broker properties as described in [Table 3–9](#).
- Click **Edit** to modify the broker properties. For details, see the [“Editing an MS SQL Server DataServer broker configuration”](#) section on page 3–32.

**Table 3–9: MS SQL Server DataServer broker properties**

(1 of 4)

Property category or property	Description
<b>General</b>	
<b>Working directory</b>	Your general working directory. The default value is @ {WorkPath}.
<b>Operating mode</b>	For DataServer brokers, the operating mode is state-aware. The default value is state-aware.
<b>Port number</b>	The type of the TCP/IP listening port that the DataServer broker uses. The default value is 3050.

**Table 3–9: MS SQL Server DataServer broker properties***(2 of 4)*

Property category or property	Description
<b>Auto start</b>	Starts the DataServer automatically when the AdminServer starts. The <b>Auto start</b> check box is cleared by default.
<b>Owner Information</b>	
<b>Group name</b>	(UNIX only) The name of the group.
<b>Username</b>	Your user name. The default is blank.
<b>Password</b>	Your password. The default is blank.
<b>Controlling NameServer</b>	
<b>Register with NameServer</b>	A checkbox that indicates whether to register the DataServer with a controlling NameServer.
<b>Controlling NameServer</b>	Identifies the NameServer that the DataServer uses to find a broker that supports the correct application services. (If you did not select to register the DataServer with a NameServer, the field is dimmed and unavailable.)
<b>Registration mode</b>	<p>How the broker specifies its hostname when registering with its controlling NameServer. This hostname information is passed onto a client application when it attempts to connect to a Data Service that the broker supports. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>Register IP</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li> <li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li> <li>• <b>Register-HostName</b> — The broker registers with the values specified in the hostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li> </ul>
<b>Registration host name</b>	(UNIX only) The name of the group.

**Table 3–9: MS SQL Server DataServer broker properties***(3 of 4)*

Property category or property	Description
<b>Data Service</b>	
<b>Data service name list</b>	Data services are services that are provided by a selected broker's servers. The default service name is the broker name. The default value is blank.
<b>Supports default service</b>	A check box that indicates whether the MS SQL Server broker supports the default service. Select the check box to have the broker support the default service, or click to clear the check box and cancel support for the default service. The <b>Supports default service</b> check box is cleared by default.
<b>Logging Setting</b>	
<b>Broker log filename</b>	The general broker log filename. The default is @{WorkPath}\mssbroker1.broker.log.
<b>Broker logging level</b>	<p>The amount of information written to the broker log. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — No log file is written.</li> <li>• <b>Error Only</b> — Only error-related information is written to the log file, which reduces the size of the log file.</li> <li>• <b>Basic</b> — Only some information is written to the log file.</li> <li>• <b>Verbose</b> — All broker activity is written to the log file. This might produce large files that use up disk space.</li> <li>• <b>Extended</b> — Information in addition to <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	Create a new broker log file each time a broker is started, even if the broker log file specified already exists. The <b>Append to broker log file</b> check box is selected by default.
<b>Broker logging entry types</b>	A single entry or comma delimited list of logging entry types.
<b>Broker log file threshold size</b>	Specifies a limit on how big the log file can get (in bytes). When the log is full, the broker creates a new log file with a sequence number.
<b>Maximum number of broker log files</b>	Limit on how many broker log files will be kept on the system when there is a threshold limit. It represents the number of log files including the current on and includes any rolled over log files that already exist when the process started.

**Table 3–9: MS SQL Server DataServer broker properties***(4 of 4)*

Property category or property	Description
<b>Advanced Features</b>	
<b>Maximum client instances</b>	<p>The maximum number of client connections that the broker can support concurrently.</p> <p>The default value of 512 that is set for this parameter is high enough to ensure that the number of client connections is unlimited. Realistically, however, the system-level resources needed to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you may need to reduce the maximum number of client connections to a more reasonable value. Note, however, that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads alleviates the problem.</p>
<b>Priority weight (0-100)</b>	An integer value between 0 and 100 that influences the share of the workload that the selected broker receives. The larger the value, the heavier the load that is distributed to the broker. Using the <b>Priority weight</b> value, the NameServer distributes client requests to the same DataServer service across all DataServer brokers. The default value is 0.
<b>Registration retry (in seconds)</b>	After registering with its controlling NameServer, the DataServer broker periodically sends "keep-alive" messages to the NameServer to let the NameServer know that the broker is still active. The <b>Registration retry</b> value is the number of seconds that pass between "keep-alive" messages. The default value is 30.

## Viewing or modifying MS SQL Server DataServer server properties

You can view or modify MS SQL Server DataServer properties in the following categories: General and Logging Setting.



### To view or modify server properties for an MS SQL Server broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **MSS DataServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the MS SQL Server server properties as described in [Table 3–10](#).
- Click **Edit** to modify the server properties. For details, see the “[Editing an MS SQL Server DataServer broker configuration](#)” section on page 3–32.
- View the MS SQL Server DataServer log file. For details, see the “[Viewing the MS SQL Server DataServer log files](#)” section on page 3–36.

**Table 3–10: MS SQL Server DataServer server properties**

(1 of 2)

Property category or property	Description
<b>General</b>	
<b>Server executable file</b>	The pathname of the DataServer executable that the broker runs. The default for the DataServer for MS SQL Server in Windows is:  @{Startup\DLC}bin\_mssqlrsrv.exe
<b>Server startup parameters</b>	The startup parameters for the server. You should not modify the default parameters, which are:  -svub -S -X -N TCP -U X -P x -hs 0 -s 40
<b>Minimum port number</b>	The minimum port number that you want to make available for each DataServer broker. The default is blank.
<b>Maximum port number</b>	The maximum port number that you want to make available for each DataServer broker. The default is blank.

**Table 3–10: MS SQL Server DataServer server properties**

(2 of 2)

Property category or property	Description
<b>Logging Setting</b>	
<b>Server log filename</b>	<p>The DataServer log filename. Enter a valid path name.</p> <p>Installed Default: @{WorkPath}\mssbroker1.server.log</p>

### Setting MS SQL Server DataServer broker environment variables

You can set environment variables for the DataServer broker for MS SQL Server. You can set one or more environment variables that:

- Are accessible from your ABL application
- Affect the environment in which your application runs

For the MS SQL Server DataServer, there are no required environment variables in Windows.



#### To set MS SQL Server DataServer environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker whose environment variables you want to edit. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.
7. Click **Save** when you finish adding the variables and their values.

### Editing an MS SQL Server DataServer broker configuration

You can edit an MS SQL Server DataServer broker configuration from the management console.



#### To edit an MS SQL Server DataServer configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker whose configuration you want to edit. The **Details** page for the broker appears in the detail frame.

3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**. (For details about the broker and server properties, see [Table 3–9](#) and [Table 3–10](#), respectively.)
5. Make the configuration changes, and then click **Save**.

## Creating an MS SQL Server DataServer broker

OpenEdge Management and OpenEdge Explorer provide a predefined default MS SQL Server DataServer configuration; however, you can create a new configuration by modifying the default settings.



### To create an MS SQL Server DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **MSS DataServer**. The **MSS DataServer Configuration** page appears.
4. Type the name of the new DataServer in the field provided.
5. Click **Save**. The **MSS DataServer Configuration** page appears, allowing you to establish the broker's properties.

## Deleting an MS SQL Server DataServer broker

You can delete an inactive DataServer broker for MS SQL Server from the management console.

If a broker is running, you cannot delete it. You must first stop the broker and then delete it.



### To delete an MS SQL Server DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker you want to delete. The **Details** page for the broker appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Starting an MS SQL Server DataServer broker

You can start an MS SQL Server DataServer broker from the management console.



### To start an MS SQL Server DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker you want to edit. The **Details** page for the broker appears in the detail frame.
3. Click **Control**. The broker's **Control** page appears.
4. Click **Start Oracle DataServer**.

## Starting an MS SQL Server DataServer broker automatically

You can set a broker for MS SQL Server to start automatically when the AdminServer starts.



### To start an MS SQL DataServer broker automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker you want to start. The **Details** page for the broker appears in the detail frame.
3. Click **Configuration**. The **MSS DataServer Configuration** page appears.
4. Click **Edit**.
5. Select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.

## Stopping an MS SQL Server DataServer broker

You can stop an MS SQL Server DataServer broker from the management console.



### To stop an MS SQL Server DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the MS SQL Server DataServer broker you want to stop. The **Details** page for the broker appears in the detail frame.



3. Click **Control**. The **MSS DataServer Control** page appears.
4. Click **Stop MSS DataServer**, and then click **OK** to confirm the deletion.

## Viewing the status of an MS SQL Server DataServer broker

You can view the status of any active MS SQL Server DataServer broker, server groups, and auxiliary processes through the management console. Most of the broker instances provide a status view, enabling you to monitor the status of multiple brokers simultaneously.

To see the status of a broker, the broker must be running or have been running.



### To view the status of an MS SQL Server DataServer broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.
3. Click **Status**. A Status summary and summary details appear and provide the information described in [Table 3–11](#) and [Table 3–12](#), respectively.

**Table 3–11: MS SQL Server DataServer broker property status summary** (1 of 2)

Property	Description
<b>Host</b>	The broker's host machine.
<b>Broker Name</b>	The name of the broker whose status you are viewing.
<b>Operating Mode</b>	For DataServer brokers, the operating mode is state-aware.
<b>Broker Status</b>	The current state of the broker.
<b>Broker Port</b>	The TCP/IP port number that the broker listens to.
<b>Broker PID</b>	The process ID of the broker.
<b>Active Servers</b>	The number of active servers.
<b>Busy Servers</b>	N/A
<b>Locked Servers</b>	N/A
<b>Available Servers</b>	N/A
<b>Active Clients (now, peak)</b>	N/A

**Table 3–11: MS SQL Server DataServer broker property status summary***(2 of 2)*

Property	Description
<b>Client Queue Path (cur, max)</b>	The number of clients waiting for agents to become available to service their request. The current value ( <b>cur</b> ) represents the number of waiting clients at the moment the status is displayed, and the maximum value ( <b>max</b> ) represents the largest number of clients waiting concurrently since the broker was started.
<b>Total Requests</b>	N/A
<b>Rq Wait (max, avg)</b>	N/A
<b>Rq Duration (max, avg)</b>	N/A

**Table 3–12: MS SQL Server DataServer broker status details**

Property	Description
<b>PID</b>	The process id of the server process
<b>State</b>	The current state of the server process
<b>Port</b>	The TCP/IP port number used by the server
<b>nRq</b>	N/A
<b>nRcvd</b>	N/A
<b>nSent</b>	N/A
<b>Started</b>	The time stamp indicating when the server process is started
<b>Last Change</b>	The time stamp indicating when the server process last changed state

## Viewing the MS SQL Server DataServer log files

You can view the log files for an MS SQL Server DataServer by using the log file viewer. The log file viewer allows you to examine the MS SQL Server DataServer log files through an HTML interface.



**To access the log file viewer:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for the broker appears in the detail frame.

### 3. Click **Log File Viewer of Broker** or **Log File Viewer of Servers**.

You can use the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display 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, regardless of 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 **Previous** 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**.
- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.

- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

---

## Configuring NameServers

---

You can configure NameServer property settings and add new NameServer instances in OpenEdge Management and OpenEdge Explorer. You can also start, stop, and delete NameServer instances, as well as view their status and log files.

This chapter provides the following information:

- [NameServer configuration and administration](#)
- [Configuring a NameServer](#)
- [Creating a NameServer](#)
- [Configuring fault tolerance and load balancing](#)
- [Starting a NameServer](#)
- [Stopping a NameServer](#)
- [Deleting a NameServer](#)
- [Viewing the status of a NameServer](#)
- [Handling UDP and firewall issues](#)
- [Viewing the NameServer log file](#)

## NameServer configuration and administration

You can optionally use the NameServer to mediate client connections for instances of OpenEdge Unified Broker products that you configure with OpenEdge Management or OpenEdge Explorer. The Unified Broker products include all OpenEdge resources that are managed through the AdminServer. For more information on the Unified Broker products and the AdminServer, see *OpenEdge Getting Started: Installation and Configuration* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site.

When you choose to use the NameServer, each component instance registers with only one NameServer, which is known as the instance's controlling NameServer. Each time an instance starts up, it registers with its controlling NameServer by sending its location and other configuration information. The NameServer uses this information to help resolve client connection requests. A particular NameServer can accept registrations from any number of instances, including instances of more than one type of component.

Depending on the way you configure and use the OpenEdge product, you can decide to specify a controlling NameServer to handle client connection requests. For example, your configuration might be simple and not require the location transparency the NameServer provides, or it might be more complex and, therefore, dependent on this and other NameServer features.

The OpenEdge Unified Broker products that you can optionally choose to register with the NameServer include:

- AppServers
- AppServer Internet Adapters
- DataServers for ODBC, Oracle, and MS SQL Server configured to use the NameServer
- SonicMQ Adapters
- WebSpeed Transaction Servers

### NameServer features

You can use OpenEdge Management or OpenEdge Explorer to configure both local NameServer instances and remote NameServer instances. The NameServer installation provides one sample NameServer (NS1), and you can use the sample as a template for creating and configuring additional NameServers.

In its role as a connection arbitrator, the NameServer provides location transparency, connection-level fault tolerance, server-level fault tolerance, and optional load balancing. For details about these features, see the [“Configuring fault tolerance and load balancing”](#) section on page 4–14.

## Working with the NameServer

To work with a NameServer instance, select the **NameServer** folder from the management console's list pane. There is one pre-defined default NameServer (NS1).

You can perform the following NameServer actions:

- [Configuring a NameServer](#)
- [Editing a NameServer configuration](#)
- [Creating a NameServer](#)
- [Starting a NameServer](#)
- [Stopping a NameServer](#)
- [Deleting a NameServer](#)
- [Configuring fault tolerance and load balancing](#)
- [Viewing the status of a NameServer](#)
- [Handling UDP and firewall issues](#)
- [Viewing the NameServer log file](#)

## NameServer as an optional client connection request handler

Configuring a NameServer to work with the AppServer, a DataServer, a SonicMQ Adapter, or a WebSpeed Transaction Server is optional. Depending on the way you configure and use the OpenEdge product, you can decide if you want to specify a controlling NameServer. For example, your configuration might be simple and not require location transparency or load balancing, or it might be more complex and, therefore, dependent on those features.

## Local and remote NameServers

Before configuring a NameServer, you must know the network location where it is to run. For any network host, you can configure two types of NameServer instances:

- **Local** — An instance that runs locally on the host where it is defined
- **Remote** — An instance that references a NameServer that is defined and runs on a machine that is remote from the host where the reference is defined

The purpose of defining a remote NameServer instance is to provide an easy way of having multiple Unified Broker products refer to a controlling NameServer that runs on another machine. You cannot start, stop, obtain status, or modify any properties of a remote NameServer instance other than the host name and port number that it references.

## Configuring a NameServer

The OpenEdge® NameServer controls a pool of brokers that register (with the NameServer) the application services they provide. The NameServer can then direct client connection requests to a broker that supports a requested application service. The NameServer also provides location transparency, and, when the NameServer Enterprise Edition is installed, the ability to perform load balancing. Load balancing allows you to balance client workload among multiple brokers that service the same application (that is, the same set of procedures and resources).

For more information, see the [“Configuring local instances”](#) section on page 4–4 and the [“Configuring remote instances”](#) section on page 4–9.

### Configuring local instances

You configure a local NameServer to run on the selected host where it is defined. Unified Broker products running on the same host as a local NameServer instance can use it as their controlling NameServer.

You can view or edit the configuration properties of a NameServer from the management console. You can also create a new NameServer, view a NameServer’s status, delete a NameServer you no longer need, or view the NameServer’s log file.



#### To configure a local NameServer instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired NameServer broker instance. The **Details** page for the instance appears in the detail frame.
3. In the **Command and control** section of the page, click **Configuration**. The **NameServer Configuration** page appears.

From this page, you can do the following:

- View a read-only display of the NameServer properties as described in [Table 4–1](#).
- Click **Edit** to modify the NameServer properties. For details, see the [“Editing a NameServer configuration”](#) section on page 4–10.
- Click **Create** to create a NameServer instance. For details, see the [“Creating a NameServer”](#) section on page 4–12.
- Click **Delete** to delete a NameServer instance. For details, see the [“Deleting a NameServer”](#) section on page 4–13.



Table 4–1: NameServer properties

(1 of 4)

Property category or property	Description
<b>Location</b>	
<b>Location</b>	<p>A read-only property that indicates whether the NameServer is local or remote:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> — The NameServer runs locally on the selected host. You can configure all the properties of a local NameServer.</li> <li>• <b>Remote</b> — The NameServer runs remotely on a network machine that is separate from the selected host.</li> </ul> <p>You can set this property only when you first create the NameServer instance. To change the location of a NameServer, you must first delete and then recreate the NameServer with the new location setting.</p>
<b>Host name</b>	The name of the host. If the NameServer is local, this is set to <b>localhost</b> . Otherwise, type the name of the remote host where the NameServer is to run.
<b>Port number</b>	The number of the UDP port that the NameServer uses to listen for client connection requests and registration messages from AppServers, DataServers, and WebSpeed Transaction Servers.
<b>General</b>	
<b>Working directory</b>	The NameServer working directory. Enter the pathname.
<b>Broker keep alive timeout</b>	<p>A value, in seconds, that indicates how often the NameServer checks for Unified Broker instances that have timed out. When an OpenEdge Unified Broker instance registers with a NameServer, the instance indicates how often it sends "keep-alive" messages by setting a registration retry value (a property setting in Advanced Features for the Unified Broker). Once a NameServer determines that it did not receive a "keep-alive" message from a Unified Broker instance within the broker's registration retry time, the NameServer automatically unregisters the instance.</p> <p>Note that you should use a <b>Broker keep alive timeout</b> value that is somewhat larger than the Unified Broker instance's registration retry value. The NameServer adjusts the specified value to allow for normal networking delays that can occur within your computer network. To keep the NameServer from using up computer resources unnecessarily, set the <b>Broker keep alive timeout</b> to a value that is at least 30% larger than the typical Unified Broker registration retry value.</p>

Table 4–1: NameServer properties

(2 of 4)

Property category or property	Description
<b>Auto start</b>	A checkbox that you can select if you want the NameServer to start automatically when the controlling AdminServer starts.
<b>Logging Setting</b>	
<b>Server log filename</b>	<p>The NameServer log filename. Enter a valid path name.</p> <p>The installed default is @{WorkPath}\NS1.NS.log.</p>
<b>NameServer logging level</b>	<p>A value that specifies the amount of information that is written to the NameServer log. Select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul> <p>The installed default is <b>Basic</b>.</p>
<b>Append to NameServer log file</b>	<p>A check box that indicates whether a new server log file should be created when the NameServer is started.</p> <p>To create a new NameServer log file each time the NameServer is started, even if the NameServer log file already exists, clear the check box.</p> <p>To append log entries to the existing NameServer log file, select the check box.</p> <p><b>Append to NameServer log file</b> is selected by default.</p>

Table 4–1: NameServer properties

(3 of 4)

Property category or property	Description
<b>Logging entry types</b>	<p>The default and only supported entry is NSPlumbing. Specifying the NSPlumbing log entry type turns on logging for different NameServer actions based on logging level:</p> <ul style="list-style-type: none"> <li>• <b>Basic</b> — Logs messages when a NameServer starts and stops, for load balancing status, and when registering and unregistering brokers. This is the default logging level.</li> <li>• <b>Verbose</b> — Same as <b>Basic</b>, plus logs requests from clients and status information on the communication between client and NameServer and between NameServer and brokers.</li> <li>• <b>Extended</b> — Same as <b>Verbose</b>, plus logs debugging messages on broker unregistering.</li> </ul> <p>Log entries generated for this type contain the tag NSPlumbing.</p>
<b>Log file threshold size</b>	<p>A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.</p>
<b>Maximum number of log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of NameServer log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p style="text-align: center;">filename.#####.extension</p>

Table 4–1:      NameServer properties

(4 of 4)

Property category or property	Description
<b>Advanced Features</b>	
<b>Neighboring NameServers</b>	<p>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). The list contains all NameServer instances defined in the management console, initially with none selected. To make a NameServer a neighboring NameServer, select the unselected NameServer to highlight it. You can select as many NameServer instances as you want to be neighboring NameServers. Each neighboring NameServer thus appears highlighted in the list. To remove a NameServer from the list of neighboring NameServers, select the already-selected NameServer to remove the highlight.</p> <p>The setting of this property is optional. When a NameServer receives a request for an application service name that is unknown and there are no neighboring NameServers specified, the NameServer sends a message to the requesting client indicating that the application service is unknown. When such a NameServer has neighboring NameServers specified, it forwards the request to each of the neighboring NameServers. The first neighboring NameServer that has the application service name, and responds to the connection request, provides the specified connection to the client.</p>
<b>TCP/IP Version</b>	<p>Internet Protocol for network communication:</p> <ul style="list-style-type: none"> <li>• <b>IPv4</b> — Accepts only IPv4 connections.</li> <li>• <b>IPv6</b> — Allows IPv4 and IPv6 connections using mapped address (where supported).</li> </ul> <p>The default is IPv4.</p> <p>You must also set the <b>jvmArgs</b> property in the <code>ubroker.properties</code> file to override the default behavior of the JVM for IPv6 connections.</p> <pre>jvmArgs=-Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true</pre>
<b>Environment variables</b>	
<b>Environment</b>	<p>Allows you to add and delete environment variables.</p> <p>Environment variables that you add to the list are scoped to the process in which the NameServer executes.</p>

## Configuring remote instances

A remote NameServer instance is a reference to a NameServer that runs on some other host. By defining a remote instance, you allow a Unified Broker running on the local host to use the NameServer running on the remote host as its controlling NameServer. Thus, a remote NameServer configuration requires that you define:

- One local NameServer instance on the host where the NameServer runs
- One remote instance on each host where a Unified Broker uses the remote NameServer as its controlling NameServer

For example, consider that you are creating a NameServer instance on one system, which is named Sys1. The NameServer instance is actually physically running under an AdminServer on another system, which is named Sys2. The instance is remote for Sys1 and local for Sys2.

To create a remote instance, you need the host name and port number of Sys2 and its NameServer. You then use that information to create a remote NameServer on Sys1.



### To configure a remote NameServer instance:

1. Configure at least one local NameServer instance on the remote host.
2. Using the management console, connect to the AdminServer on a host where you want to define the remote instance, and expand the **NameServer** folder for that host.
3. Select an existing remote NameServer instance. The **Details** page appears in the **Details** frame.

You can also create a new NameServer, specifying its name and indicating that its location is remote. For details, see the [“Creating a NameServer”](#) section on page 4–12.

4. In the **Command and control** section of the Details page, click **Configuration**. The **NameServer Configuration** page appears.
5. Click **Edit**. The **Location** default is **local**. If you are editing an existing instance or creating a new instance, change the location to **remote**.
6. Set or change the host name to a machine where you have a NameServer defined as a local instance. Set the port number to the port on which the local instance you want to reference is listening.
7. Click **Save**.

## Editing a NameServer configuration

You can edit a NameServer configuration from the management console.



### To edit a NameServer configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance whose configuration you want to edit. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The instance's **Configuration** page appears.
4. Click **Edit**. (For details about the NameServer properties, see [Table 4–1](#).)
5. Make the configuration changes, and then click **Save**. Any changes you make are saved in the `ubroker.properties` file.

## Setting or deleting NameServer environment variables

You can set or delete environment variables for a NameServer.



### To set NameServer environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance whose environment variables you want to set. The **Details** page for that NameServer appears in the detail frame.
3. Click **Configuration**. The **NameServer Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.
7. Click **Save** when you finish adding the variables and their values.



### To delete NameServer environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance whose environment variable (or variables) you want to delete. The **Details** page appears in the detail frame.
3. Click **Configuration**. The **NameServer Configuration** page appears.
4. Click **Edit**.

5. Click **Environment Variables**.
6. Delete the variable and value pair (or pairs).
7. Click **Save** when you finish.

## Creating a NameServer

OpenEdge Management and OpenEdge Explorer provide a sample configuration NameServer instance (NS1) that it uses as a default.

You can create two types of NameServer instance:

- **Local** — Where the instance definition resides on, and the NameServer itself executes on, the selected host.
- **Remote** — Where the instance definition is a reference to a NameServer that resides and executes on a machine that is remote from the selected host.



### To create a NameServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **NameServer**. The **NameServer Configuration** page appears.
4. Type the name of the new NameServer in the field provided.

---

**Note:** The NameServer name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured NameServer names.

---

5. Click **Save**. The **NameServer Configuration** page appears, allowing you to configure the NameServer's properties. (For details about the properties, see [Table 4–1](#).)

Once you create a NameServer instance, you cannot change the location setting unless you delete and recreate the instance.

---

**Note:** If you are defining a remote NameServer instance, set its host name and port number properties to reference a NameServer that is defined as a local NameServer on some other machine in your network. For details, see the “[Configuring remote instances](#)” section on page 4–9.

---



## Deleting a NameServer

You can delete any NameServer instance from the management console as long as neither of the following conditions is true:

- The NameServer instance is running.
- One or more Unified Broker instances reference this NameServer instance as their controlling NameServer.

To delete the NameServer, you must stop the NameServer instance and delete all Unified Broker instances that reference it.



### To delete a NameServer instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance you want to delete. The **Details** page for that instance appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Configuring fault tolerance and load balancing

With the NameServer, you have access to connection-level fault tolerance, server-level fault tolerance, and load balancing.

### Connection-level fault tolerance

Multiple NameServers can cooperate to resolve a client connection. If one NameServer becomes unavailable, another can take its place, thus providing connection-level fault tolerance. When you configure and use NameServers with connection-level fault tolerance, each requesting client uses the Unified Broker connection provided by the first responding NameServer and ignores the rest.

You can configure two types of connection-level fault tolerance:

- NameServer replication
- NameServer neighbors

#### NameServer replication

To use NameServer replication, you first configure multiple NameServers on one subnet to listen on the same UDP port. You then have each Unified Broker instance register with its controlling NameServer and have each client contact its connecting NameServer using the subnet broadcast address and the same UDP port number. Thus, Unified Broker registrations are automatically replicated in each NameServer, and client connection requests are handled by each NameServer. The first replicated NameServer to respond to a client request provides the Unified Broker connection.

#### NameServer neighbors

To use NameServer neighbors, you configure multiple NameServers on one or more subnets. For each such NameServer, you specify the instance name of each of the other NameServers that you want to act as a NameServer neighbor. When a client makes a connection request to a specific NameServer, if this NameServer determines that it cannot satisfy the request (usually because it does not recognize the specified application service), it passes the request to all of its NameServer neighbors. The first NameServer neighbor to respond to a client request provides the Unified Broker connection.

You can configure and use NameServer replication and NameServer neighbors together to provide the maximum in connection-level fault tolerance.

### Server-level fault tolerance

If you have the load-balancing option installed, you can configure multiple Unified Broker instances of a given type to support the same application service and share the same controlling NameServer. In this configuration, the controlling NameServer distributes client connection requests randomly across Unified Broker instances. If one Unified Broker instance becomes unavailable to handle client requests, another is available to take its place, thus providing server-level fault tolerance.

## Load balancing

If you specify a priority weight factor for each Unified Broker instance, the NameServer distributes the connection requests proportionately, thus balancing the load. For example, suppose that an Inventory service is supported by WebSpeed brokers WS1, WS2, and WS3, which have weight factors of 20, 20, and 60, respectively. The selection algorithm used by the NameServer guarantees that WS1 and WS2 are selected 20% of the time and WS3 is selected 60% of the time.

---

**Note:** You do not assign application services or weight factors to NameServers, but rather to Unified Broker instances. Each Unified Broker instance passes its weight factor and list of application service names to its controlling NameServer when it registers with the NameServer.

---

## Location transparency

When a Unified Broker instance is started, it registers with its controlling NameServer the application services that it supports. (An application service is an arbitrary designation for the business function that a Unified Broker instance provides.) The NameServer maintains a separate list of application services for each Unified Broker type. When a client of a certain type requests a connection to an application service, the NameServer redirects the connection request to a Unified Broker instance of the corresponding type that supports the requested application service. Because the client always connects indirectly to a Unified Broker through the NameServer, the client does not need to know the location of the Unified Broker instance. It needs to know only the location of the NameServer and the name of a valid application service that it wants.

## Starting a NameServer

From the management console, you can directly start a local NameServer on the machine where it runs. You can also set a NameServer to start automatically when the AdminServer starts. For details, see the “[Starting a NameServer automatically](#)” section on page 4–16.

---

**Note:** To start a remote NameServer, access the corresponding local NameServer instance selected in the container of the host on which it runs.

---



### To start a NameServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance you want to start. The **Details** page for that instance appears in the detail frame.
3. Click **Control**. The instance’s **Control** page appears.
4. Click **Start NameServer**.

## Starting a NameServer automatically

You can set a NameServer to start automatically when the AdminServer starts.



### To start a NameServer automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance you want to start automatically. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The **NameServer Configuration** page appears.
4. Click **Edit**.
5. Click the **General** tab.
6. Select the **Auto start** option by clicking in the checkbox.
7. Click **Save**.

## Stopping a NameServer

From OpenEdge Explorer, you can directly stop a local NameServer on the machine where it runs.

---

**Note:** To stop a remote NameServer, access the corresponding local NameServer instance selected in the folder of the host where it runs.

---



### To stop a NameServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the NameServer instance you want to stop. The **Details** page for that instance appears in the detail frame.
3. Click **Control**. The **NameServer Control** page appears.
4. Click **Stop NameServer**.

## Viewing the status of a NameServer

You can view the status of any running NameServer instance through the management console.

Each NameServer instance provides a separate status view. You can monitor the status of only those NameServer instances running locally on a selected host.



### To view the status of a NameServer instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired NameServer instance. The **Details** page for that instance appears in the detail frame.
3. Under **Operations views**, click **Status**. A status summary and summary details appear.

The summary, as described in [Table 4–2](#), provides collective property and statistics information for the NameServer and the OpenEdge Unified Broker instances that it coordinates.

The details, as described in [Table 4–3](#), provide a table of status information for all Unified Broker instances that are registered with the NameServer. Each row in the table contains information for a single Unified Broker instance.

**Table 4–2:      NameServer status summary properties**

Property or statistic	Description
<b>Host</b>	The name of the host machine
<b>Port</b>	The UDP port number that the NameServer listens to
<b>Timeout</b>	The Unified Broker keep alive timeout value for the NameServer
<b>Start time</b>	When the NameServer started
<b>Number of AppServices</b>	The number of application services that the NameServer is coordinating
<b>Number of Brokers</b>	The number of Unified Broker instances the NameServer is coordinating
<b>Total client requests received</b>	The number of client requests the NameServer has received
<b>Total client requests rejected</b>	The number of client requests the NameServer has rejected

**Table 4–3:      NameServer status detail properties**

<b>Property</b>	<b>Description</b>
<b>AppService Name</b>	The name of an application service registered with the NameServer.
<b>UUID</b>	The unique number for the Unified Broker instance.
<b>Name</b>	The Unified Broker instance name.
<b>Host</b>	The name of the host where the Unified Broker instance resides.
<b>Port</b>	The TCP/IP port number used by the Unified Broker instance to handle client connection requests.
<b>Weight</b>	The priority weight assigned to the Unified Broker instance for load balancing.
<b>Timeout</b>	The number of seconds that must pass between keep-alive messages sent by the Unified Broker instance before the NameServer assumes that the instance is no longer available. Once the timeout expires, the NameServer automatically unregisters the Unified Broker instance.

## Handling UDP and firewall issues

Whenever an AppServer Internet Adapter (AIA) or WebSpeed Messenger instance communicates with a NameServer, the instance is assigned a UDP port. The NameServer uses this port to send a response message to the instance. By default this port is dynamically assigned by the operating system within the UDP port range of 1024 to 65535.

If there is a firewall between the instance and the NameServer, the default setting requires that all UDP ports in that range must be open from the network where the NameServer exists to the network where the AIA or Messenger instance exists. Otherwise, communication between the instance and the NameServer cannot occur.

However, the only port that must be open in the firewall is the controlling NameServer port for UDP traffic going from the network containing the AIA or Messenger instance to the network containing the NameServer. To increase security, you can limit the range of ports from which an AIA or Messenger instance can be assigned an UDP port. The NameServer Client Port Minimum (minNSClientPort) and the NameServer Client Port Maximum (maxNSClientPort) properties can be used to reduce the number of UDP ports that you must open in the firewall. When you set these properties, OpenEdge chooses the first port in that range that is not currently used by some other application for UDP broadcast communications. If no port in this range is available, the default behavior is to retry the ports for a period of time.



## Viewing the NameServer log file

The NameServer log file viewer allows you to examine the contents of log files through an HTML interface.

For details, see the section about using the NameServer log file viewer in *OpenEdge Management: Servers, DataServers, Messengers, and Adapters*.



## **Configuring AppServers and AppServer Internet Adapters**

---

You can use OpenEdge Management or OpenEdge Explorer to configure the properties of OpenEdge resources that you are monitoring, such as AppServers or AppServer Internet Adapters. Any configuration changes that you make to either of these resources in OpenEdge Management or OpenEdge Explorer are automatically reflected in the configuration property file `ubroker.properties`.

This chapter provides the following information about working with AppServer and AppServer Internet Adapter configuration properties:

- [AppServer configuration and administration](#)
- [For additional AppServer information](#)
- [AppServer Internet Adapter configuration and administration](#)

## AppServer configuration and administration

With the AppServer, you can make ABL applications available to multiple machines over a network.

You can use OpenEdge Management or OpenEdge Explorer to configure new AppServer instances and administer the configurations of existing instances. The AppServer installation provides one sample AppServer (asbroker1). You can use this sample component as a template for creating and configuring additional AppServers. Each AppServer is called an instance.

### Understanding AppServer configuration

Each AppServer consists of an AppServer broker and one or more AppServer agents. An AppServer agent is a single ABL Virtual Machine (AVM) instance running on an AppServer. Each AppServer agent executes ABL procedures on behalf of AppServer clients, including ABL and Open Clients (Java, .NET, and Web services clients).

An AppServer can support clients using two different session models, which are dependent on its operating mode:

- **Session-Managed** — Includes stateless, state-aware, and state-reset operating modes. These operating modes maintain a single session and connection between the AppServer and the client for all client requests.
- **Session-Free** — Includes only the state-free operating mode. This operating mode maintains no session or connection between the AppServer and the client. In this operating mode, the AppServer treats any request from a client as independent from all other requests. As a result, some SESSION handle attributes in the ABL application have no valid meaning, or they provide different information.

The AppServer supports these session models for supported AppServer clients.

For more information, see *OpenEdge Application Server: Developing AppServer Applications* on the Progress Software Developers Network (PSDN) at <http://www.psdn.com/library/kbcategory.jspa?categoryID=2139>.

### AppServer features

The AppServer broker does the following:

- Optionally registers with a NameServer the application services that it provides for access by clients
- Manages connections between clients and a pool of AppServer agents
- Maintains the status of each AppServer agent in its pool and dynamically scales the number of servers according to changing demand
- When configured for stateless or state-free operating mode, dispatches remote requests to AppServer agents
- When configured for state-reset or state-aware operating mode, connects each client to a dedicated AppServer agent that handles all remote requests for that client until the connection is terminated

## Working with the AppServer

To work with an AppServer instance, double-click the **AppServer** folder in the management console's list frame. There is one pre-defined default AppServer (asbroker1).

You can perform the following AppServer actions:

- [Configuring an AppServer](#)
- [Creating an AppServer](#)
- [Starting and stopping an AppServer](#)
- [Adding or trimming AppServer agents](#)
- [Viewing the status of an AppServer](#)
- [Deleting an AppServer](#)
- [Viewing the AppServer log files](#)

## AppServer configuration properties

An AppServer starts up by using the configuration information stored in the `ubroker.properties` file. The file contains the property settings for all server resources on the host where OpenEdge is installed.

### The `ubroker.properties` file

The `ubroker.properties` file consists of a hierarchical structure of configuration entities, in which parent entities provide configuration information that you can override or extend in each child entity. Each configuration entity has a name that begins the entity definition, and the definition contains configuration settings for one or more product instances. For example, property settings for an instance of an AppServer are determined hierarchically, as shown in [Table 5–1](#).

**Table 5–1: Ubroker.properties file structure**

Configuration entity name	Configuration entity function
[UBroker]	Defines default property settings for various OpenEdge brokers, including all AppServer brokers.
[UBroker.AS]	Defines default property settings for all instances of an AppServer.
[UBroker.AS. <i>product-instance-name</i> ]	Defines property settings for this instance of an AppServer. The <code>ubroker.properties</code> file can contain several of these entities, each with a unique <i>product-instance-name</i> .

Parent entities provide default values for all of their child entities. However, at any child level a redefinition of any value supersedes the default value of its parent. All children, from their definition level down, inherit this new value.

Like the `ubroker.properties` file, a similar file, `conmgr.properties`, stores all the properties for each instance of an OpenEdge database. For more information about OpenEdge Management and OpenEdge Explorer and the database configuration properties, see [Chapter 2, “Configuring OpenEdge Databases.”](#)

The `ubroker.properties` file resides in the `properties` subdirectory of the OpenEdge install directory. The file is editable and any changes that you make to it are automatically reflected in the management console.

---

**Note:** Although making manual edits to the `ubroker.properties` file is possible, Progress Software recommends that you use the Mergeprop utility or the management console to make property changes. For more information about the Mergeprop utility, see *OpenEdge Getting Started: Installation and Configuration*.

---

## Configuring an AppServer

From the management console, you can view or modify an AppServer broker’s default configuration properties in the following overall categories:

- Broker
- Agent
- SSL
- Messaging
- Environment variables

You can also create a new AppServer instance, view an instance’s status, delete an instance you no longer need, or view the instance’s log files.

Each of these tasks is described in the following sections.

### Viewing or modifying AppServer broker properties

You can view or modify AppServer broker properties in the following categories: general properties, owner information, controlling NameServer information, AppService Name List information, logging setting details, and advanced features.



#### To view or modify AppServer broker properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer** category, and select an AppServer broker. The **Details** page for that broker appears in the detail frame.

4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Configuration** page opens with the **Broker** properties in focus.

From this page, you can do the following:

- View a read-only display of the AppServer broker properties as described in [Table 5–2](#) through [Table 5–7](#). All of these tables follow this procedure.
- Click **Edit** to modify the broker properties,. For details, see the “[Editing an AppServer configuration](#)” section on page 5–21.

**Table 5–2: AppServer Broker General properties**

Property	Description
<b>Auto start</b>	Indicates whether the server automatically starts when the controlling AdminServer starts.
<b>Port number</b>	The number of the TCP/IP port that the server broker listens on.
<b>Working directory</b>	Your working directory. The default is the working directory set during OpenEdge installation.
<b>Operating mode</b>	The operating mode for this server.

**Table 5–3: AppServer Broker Owner Information properties**

Property	Description
<b>Username</b>	The username of an account that has system-administrative rights.
<b>Group name</b>	(UNIX only) The name of the group.
<b>Password</b>	The username account password in Windows.
<b>Password confirm</b>	Confirmation of the username account password. This field appears only if you are modifying the properties.  Note that this field does not appear if the AdminServer is running on a non-Windows host machine.

**Table 5–4: AppServer Broker Controlling NameServer properties** (1 of 2)

Property	Description
<b>Register with NameServer</b>	Indicates whether or not to register the AppServer with a controlling NameServer.
<b>Controlling NameServer</b>	The name of the NameServer with which this AppServer registered. If you did not select to register the broker with a NameServer, the field is disabled.

**Table 5–4: AppServer Broker Controlling NameServer properties** (2 of 2)

Property	Description
<b>Registration mode</b>	<p>How the broker specifies its hostname if it is registering with a controlling NameServer.</p> <p>This hostname information is passed onto a client application when it attempts to connect to an application service that the AppServer broker supports. The choices are:</p> <ul style="list-style-type: none"><li>• <b>Register- IP</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li><li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li><li>• <b>Register-HostName</b> — The broker registers with the values specified in the HostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li></ul>
<b>Registration host name</b>	The host name alias of the local system.

**Table 5–5: AppServer Broker AppService Name List properties**

Property	Description
<b>Application service names</b>	A list to which you can add or delete application service names.
<b>Supports default service</b>	Indicates whether or not the AppServer broker supports the default service.



Table 5–6: AppServer Broker Logging Setting properties

(1 of 3)

Property	Description
Broker log filename	The broker log filename.
Broker logging level	<p>A value that specifies the amount of information that is written to the broker log. The following values are available:</p> <ul style="list-style-type: none"><li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li><li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li><li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li><li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li><li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li></ul>
Append to broker log file	Indicates whether or not a new broker log file should be created when the AppServer is started.

**Table 5–6: AppServer Broker Logging Setting properties***(2 of 3)*

Property	Description
<b>Broker logging entry types</b>	<p>A comma-separated list of the following valid entry types for agent logging:</p> <ul style="list-style-type: none"> <li>• <b>UBroker.AutoTrim</b> — Logs messages about automatic agent thread trimming by the broker based on the <b>Auto trim timeout</b> property. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.Basic</b> — Logs messages about broker and agent startup and shutdown at the following logging levels: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs NameServer registration and connections from clients.</li> <li>– <b>Verbose</b> — Logs information about all the property values read from the <code>ubroker.properties</code> file for that broker instance, and more information on the connection from clients.</li> </ul> </li> <li>• <b>UBroker.ThreadPool</b> — Logs messages about the pool of threads managed by the broker. These threads are used to control the client requests and the agent processes. This entry type logs messages when adding, removing, and communicating with these threads. Information is logged at logging level <b>Basic</b> and higher.</li> </ul> <p>Turn on the following logging entry types only at the request of Progress Software Corporation Technical Support:</p> <ul style="list-style-type: none"> <li>• <b>UBroker.ClientFSM</b> — Logs messages about the state of the client thread as it processes the requests, from the moment it receives a connection request to the end of the request. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerFSM</b> — Logs messages about the state of the agent thread as it processes the requests, from the moment it receives a connection request to the end of the request. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMsgStream</b> — Logs messages about messages exchanged between the broker and client threads in binary format. Information is logged at logging level <b>Basic</b> and higher.</li> </ul>

Table 5–6: AppServer Broker Logging Setting properties

(3 of 3)

Property	Description
<b>Broker logging entry types</b> <i>(continued)</i>	<ul style="list-style-type: none"> <li>• <b>UBroker.ServerMsgStream</b> — Logs messages about messages exchanged between the broker and agent threads in binary format. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMsgQueue</b> — Logs messages about the client thread as it processes queued messages. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerMsgQueue</b> — Logs messages about the agent thread as it processes queued messages. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMemTrace</b> — Traces messages being processed by the client threads. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerMemTrace</b> — Traces messages being processed by the agent threads. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.Stats</b> — Logs statistics about the number of requests, plus maximum, minimum, and average request wait and duration times. Information is logged at logging level <b>Basic</b> and higher.</li> <li>• <b>UBroker.All</b> — Logs all Unified Broker entry types.</li> </ul> <p>You can specify a separate logging level for each log entry type and use a wildcard (*) to specify multiple entry types. For example:</p> <pre>UBroker.Basic:3,UBroker.ThreadPool:4,UBroker.Client*</pre> <p>For details, see the information about client logging in <i>OpenEdge Deployment: Startup Command and Parameter Reference</i>.</p>
<b>Broker log file threshold size</b>	<p>A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.</p>
<b>Maximum number of broker log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of broker log files to keep. The specified number represents the maximum number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Broker log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <pre>filename.#####.extension</pre>

**Table 5–7: AppServer Broker Advanced Features properties***(1 of 2)*

Property	Description
<b>Maximum client instances</b>	<p>The maximum number of client connections that the broker can support concurrently.</p> <p>The default is 512, which is high enough to ensure that the number of client connections is virtually unlimited. Realistically, however, the system-level resources needed to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you might need to reduce the maximum number of client connections to a more reasonable value.</p> <p>Note, however, that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads alleviates the problem.</p>
<b>Priority weight (0-100)</b>	<p>An integer value between 0 and 100 that influences the share of the workload that the selected AppServer receives. The larger the value, the heavier the load that is distributed to the server. The NameServer distributes client requests across all AppServers that have the same application service in proportion to the <b>Priority weight</b> value. The default is 0.</p>
<b>Registration retry (in seconds)</b>	<p>After registering with its controlling NameServer, the AppServer periodically sends "keep-alive" messages to the NameServer to let the NameServer know that it is still active. The <b>Registration retry</b> value is the number of seconds that pass between "keep-alive" messages.</p> <p>The default is 30.</p>
<b>Server startup timeout</b>	<p>Specifies the amount of time, in seconds, that the broker waits for an active agent to become available before starting a new agent process.</p> <p>The default is 3.</p>
<b>Request timeout</b>	<p>Specifies the amount of time, in seconds, that the broker will wait for an agent to become available for processing a request.</p> <p>The broker waits for this period only if the <b>Maximum servers</b> setting has been reached. After the timeout has expired, the client receives a "no servers available" error message.</p> <p>The default is 15.</p>

**Table 5–7: AppServer Broker Advanced Features properties** (2 of 2)

Property	Description
<b>Auto trim timeout</b>	<p>The amount of time, in seconds, that the broker waits before automatically trimming the number of running agents.</p> <p>The broker keeps track of the maximum number of agents that are simultaneously busy during the interval you specify. At the end of the interval, the broker attempts to trim the number of agents to match the <b>Maximum servers</b> for the interval, or the <b>Minimum servers</b>, whichever is greater.</p> <p>You can disable this feature by setting the timeout to zero (0). The default <b>Server Auto-trim Interval</b> is 1800 seconds.</p> <p>Progress Software recommends that you use this default setting to avoid unnecessary process management that might result from frequent starting and trimming of agents.</p>
<b>TCP/IP version</b>	<p>Internet Protocol for network communication:</p> <ul style="list-style-type: none"> <li>• <b>IPv4</b> — Accepts only IPv4 connections.</li> <li>• <b>IPv6</b> — Allows IPv4 and IPv6 connections using mapped address (where supported).</li> </ul> <p>The default is IPv4.</p> <p>You must also set the <code>jvmArgs</code> property in the <code>ubroker.properties</code> file to override the default behavior of the JVM for IPv6 connections.</p> <pre>jvmArgs=-Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true.</pre>
<b>jvmargs</b>	Java System Properties.

### Viewing or modifying AppServer Agent properties

You can modify the following Agent properties for the AppServer Broker: general properties, logging setting details, pool range, and advanced features.



#### To view or modify AppServer Agent properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer** category, and select an AppServer broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Configuration** page opens with the **Broker** properties in focus.

5. Click the **Agent** tab.

From this page, you can do the following:

- View a read-only display of the AppServer Agent properties as described in [Table 5–8](#) through [Table 5–11](#). All of these tables follow this procedure.
- Click **Edit** to modify the broker properties. For details, see the “[Editing an AppServer configuration](#)” section on page 5–21.

**Table 5–8: AppServer Agent General properties**

Property	Description
<b>Server executable file</b>	Either the default agent executable pathname or a different agent executable file of your choosing. You generally need to specify a nondefault value only if you generated a new executable using the OEBUILD utility.
<b>Server startup parameters</b>	The OpenEdge startup parameters that you want to specify to start each agent.
<b>PROPATH</b>	The search path that agents use to locate ABL procedures that they execute. Specify a list of directory pathnames specified in the form of an OpenEdge PROPATH. Make sure that you copy all of your AppServer procedures to one or more of the directories in the PROPATH. You can enter the pathname directly.
<b>Minimum port number</b>	The minimum TCP/IP port number from a range that each agent process can listen on. When each agent starts, it allocates a port that is not being used from the specified range up to the <b>Maximum port number</b> setting.
<b>Maximum port number</b>	The maximum TCP/IP port number from a range that each agent can listen on. When each agent starts, it allocates a port that is not being used from the specified range down to the <b>Minimum port number</b> setting.

**Table 5–9: AppServer Agent Logging Setting properties***(1 of 4)*

Property	Description
Server log filename	The log file name.
Server logging level	<p>A value that specifies the amount of information to be written to the agent log. Each logging level name has the indicated numeric value. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"><li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li><li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li><li>• <b>Basic</b> — Server logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li><li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li><li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li></ul>
Append to server log file	Indicates whether or not a new agent log file should be created when the AppServer is started.

Table 5–9: AppServer Agent Logging Setting properties

(2 of 4)

Property	Description
Server logging entry types	<p>A comma-separated list of valid entry types for agent logging:</p> <ul style="list-style-type: none"> <li>• <b>4GLTrace</b> — Turns on the logging for the execution of the following ABL statements: RUN, FUNCTION (user-defined functions), PUBLISH and SUBSCRIBE. Logs information at <b>Basic</b> or higher.</li> <li>• <b>ASDefault</b> — Combines the ASPlumbing and DB.Connects log entry types. This is the default value for AppServer agents.</li> <li>• <b>ASPlumbing</b> — Turns on logging for different actions depending on the logging level specified: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs messages when an AppServer agent starts and stops, and when client processes connect and disconnect from the AppServer agent.</li> <li>– <b>Verbose</b> — Same as <b>Basic</b>, plus logging of the execution of the top level procedure (that is, the procedure that the client process asks the AppServer agent to execute on its behalf, but not any procedure that the top level procedure executes).</li> <li>– <b>Extended</b> — Same as <b>Verbose</b>, plus messages about agent state, and status messages about read and write socket operations. Do not turn this on unless directed to do so by Progress Software Technical Support.</li> </ul> </li> <li>• <b>DB.Connects</b> — Turns on logging of database connections and disconnections. The messages include database name and user ID number. Logs information at logging level <b>Basic</b> or higher.</li> <li>• <b>DynObjects.DB</b> — Turns on logging for dynamic database-related objects (for example, TEMP-TABLE or DATASET) as they are created and destroyed. Logs information at <b>Basic</b> or <b>Verbose</b>.</li> <li>• <b>DynObjects.XML</b> — Turns on logging for dynamic XML-related objects (for example, x-document or x-noderef) as they are created and destroyed. Logs information at level <b>Basic</b> or <b>Verbose</b>.</li> </ul>



**Table 5–9: AppServer Agent Logging Setting properties**

(3 of 4)

Property	Description
<b>Server logging entry types</b> <i>(continued)</i>	<ul style="list-style-type: none"> <li>• <b>DynObjects.Other</b> — Turns on logging for dynamic objects that do not match the other dynamic object categories (for example, procedure or server socket) as they are created and destroyed. Logs information at <b>Basic</b> or <b>Verbose</b>.</li> <li>• <b>ProEvents.UI.Char</b> — Turns on logging of keystroke events for printable characters. Logs information at <b>Basic</b> or higher.</li> <li>• <b>ProEvents.UI.Command</b> — Turns on logging of keystroke events for nonprintable characters at the following logging levels: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs nonprintable keystrokes and a subset of Windows GUI events.</li> <li>– <b>Verbose</b> — Logs nonprintable keystrokes and all Windows GUI events.</li> </ul> </li> <li>• <b>ProEvents.Other</b> — Turns on logging of COM, asynchronous, and server socket events. Logs information at <b>Basic</b> or higher, beginning with the events for which you have written triggers.</li> <li>• <b>QryInfo</b> — Turns on database query information logging at the following logging levels, with the same amount of information logged for each level: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs static queries and initial dynamic queries.</li> <li>– <b>Verbose</b> — Logs static queries and multiple instances of dynamic queries.</li> </ul> </li> <li>• <b>SAX</b> — Turns on logging for various stages of execution using the ABL SAX parser.</li> </ul> <p>You can specify a separate logging level for each log entry type and use a wildcard (*) to specify multiple entry types. For example:</p> <p style="padding-left: 40px;">4GLTrace:3, ASPlumbing:2,DynObjects.*</p> <p>For details, see the information about client logging in <i>OpenEdge Deployment: Startup Command and Parameter Reference</i>.</p>
<b>Server log file threshold size</b>	A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the agent log file size other than what the operating system imposes.

**Table 5–9: AppServer Agent Logging Setting properties**

(4 of 4)

Property	Description
<b>Maximum number of server log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of agent log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Server log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p><i>filename.#####.extension</i></p>
<b>Server watch dog interval</b>	The interval (in seconds) for the server log file watchdog thread. If the <b>Server log file threshold size</b> property is specified, the broker will roll over the server log file when its size is larger than the logThreshold value.

**Table 5–10: AppServer Agent Pool Range properties**

Property	Description
<b>Initial number of servers to start</b>	The number of agents you want the broker to start initially.
<b>Minimum servers</b>	The minimum number of agents before the broker starts additional agents. If you trim the number of agents below this value, the server starts any additional agents needed to maintain the specified minimum when the next client connects.
<b>Maximum servers</b>	The maximum number of agents that this server can have running at the same time.

**Table 5–11: AppServer Agent Advanced Features properties**

(1 of 2)

Property	Description
<b>4GL debugger enabled</b>	Indicates if the AppServer remote debugging facility is enabled. If the facility is enabled, ABL client applications have the ability to step into remote AppServer procedures and WebSpeed procedures have the ability to step into remote AppServer procedures.
<b>Activate</b>	The name of an ABL procedure that executes before the AppServer executes a remote procedure request for a client connection that is unbound. The <b>Activate</b> procedure is available only for an AppServer running in stateless operating mode.

**Table 5–11: AppServer Agent Advanced Features properties** (2 of 2)

Property	Description
<b>Deactivate</b>	The name of an ABL procedure that executes after the AppServer executes a remote procedure request for a client connection that is unbound. The <b>Deactivate</b> procedure is available only for an AppServer running in stateless operating mode.
<b>Connect</b>	The name of an ABL procedure that executes when a client attempts to connect to this AppServer.
<b>Disconnect</b>	The name of an ABL procedure that executes when a client disconnects from this AppServer.
<b>Startup</b>	The name of an ABL procedure that executes when an AppServer agent starts up for this AppServer. The <b>Startup</b> procedure is not available for an AppServer running in state-reset operating mode.
<b>Shutdown</b>	The name of an ABL procedure that executes in an AppServer agent when the AppServer agent shuts down. AppServer agents shut down when you use the trim feature and also when you shut down the AppServer. The <b>Shutdown</b> procedure is not available for an AppServer running in state-reset operating mode.
<b>Startup parameters</b>	<p>The parameters to pass the <b>Startup</b> procedure when it executes. The <b>Startup</b> procedure takes a character string as an input parameter. For example:</p> <pre>DEFINE INPUT PARAMETER startup-data AS CHARACTER NO-UNDO.</pre> <p>You can set this parameter to any arbitrary value. If you do not specify a parameter in this field, the parameter is set to the unknown value (?) when the AppServer agent executes the <b>Startup</b> procedure.</p>

### Viewing or modifying AppServer SSL properties

You can view or modify the following SSL properties for an AppServer Broker: general properties and advanced features.



#### To view or modify AppServer SSL properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer** category, and select an AppServer broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Configuration** page opens with the **Broker** properties in focus.
5. Click the **SSL** tab.

From this page, you can do the following:

- View a read-only display of the AppServer SSL properties as described in [Table 5–12](#) and [Table 5–13](#). These tables follow this procedure.
- Click **Edit** to modify the SSL properties. For details, see the “[Editing an AppServer configuration](#)” section on page 5–21.

**Table 5–12: AppServer SSL General properties**

Property	Description
<b>Enable SSL client connections</b>	If checked, specifies that all connections to this AppServer must use SSL tunneling.
<b>Private key/Digital certificate alias name</b>	The alias name within the OpenEdge keystore of the private key and digital certificate entry to use to authenticate all connections to this AppServer. If not otherwise selected, the AppServer uses the default_server server certificate alias.
<b>Password to access the key/certificate</b>	<p>The password to use for accessing the private key and digital certificate.</p> <p><b>Note:</b> OpenEdge Explorer encrypts your password in the <code>ubroker.properties</code> file. If you set the password in <code>ubroker.properties</code> manually, you must specify the password as an encrypted value. You can obtain this value using the <code>genpassword</code> utility located in the <code>bin</code> directory of your OpenEdge installation.</p> <p>If you use the default_server server certificate, it also has a default password that you do not need to specify.</p>
<b>Password confirm</b>	Confirm the password. This field appears only if you are editing the properties.

**Table 5–13: AppServer SSL Advanced Features properties**

Property	Description
<b>Disable SSL session caching</b>	If this property is checked, caching for the SSL client session is disabled.
<b>SSL session cache timeout</b>	<p>The length of time, in seconds, that an SSL client session is held in the session cache, during which an SSL client can resume its session.</p> <p>The default is 180.</p>

## Viewing or modifying AppServer Messaging properties

You can view or modify several AppServer Messaging properties.



### To view or modify AppServer Messaging properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer** category, and select an AppServer broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Configuration** page opens with the **Broker** properties in focus.
5. Click the **Messaging** tab.

From this page, you can do the following:

- View a read-only display of the AppServer Messaging properties as described in [Table 5–14](#).
- Click **Edit** to modify the AppServer Messaging properties. For details, see the [“Editing an AppServer configuration”](#) section on page 5–21.

**Table 5–14: AppServer Messaging properties**

(1 of 2)

Property	Description
<b>SonicMQ ServerConnect enabled</b>	Allows the asbroker or wsbroker agent to start a personal SonicMQ Adapter instead of going to a separate one running elsewhere.
<b>SonicMQ ServerConnect broker log filename</b>	A valid path and file name.

**Table 5-14: AppServer Messaging properties**

(2 of 2)

Property	Description
<b>Broker logging level</b>	<p>A value that specifies the amount of information to be written to the SonicMQ ServerConnect broker log. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	Indicates whether a new broker log file should be created, even if the broker log file already exists, when the SonicMQ ServerConnect broker is started.
<b>SonicMQ ServerConnect server log filename</b>	A valid path and file name.
<b>Server logging level</b>	<p>A value that specifies the amount of information to be written to the SonicMQ ServerConnect server log. Each logging level name has the indicated numeric value. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Append to server log file</b>	Indicates whether a new server log file should be created, even if the broker log file already exists, when the SonicMQ ServerConnect instance is started.

## Setting or deleting AppServer broker environment variables

You can set or delete environment variables. For the AppServer, environment variables that appear in the list are scoped to both:

- The process in which the AppServer broker executes
- Every AppServer agent that the AppServer broker starts



### To set or delete AppServer broker environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer broker whose environment variables you want to set. The **Details** page for that broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.  
  
Alternatively, you can remove an existing variable by deleting the name and value pair.
7. Click **Save** when you finish setting (or deleting) the variables and their values.

## Editing an AppServer configuration

You can edit the configuration properties for an AppServer resource from the management console.



### To edit AppServer configuration properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer** category, and select an AppServer broker. The details page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Configuration** page opens.
5. Click **Edit**, and make the changes you want to the properties. (For details about the properties, see [Table 5-2](#) through [Table 5-14](#).)
6. Click **Save**. Any changes you make are also reflected in the `ubroker.properties` file.

## Creating an AppServer

OpenEdge Management and OpenEdge Explorer provide a sample AppServer instance (asbroker1) with a default configuration. Each new instance you create uses the same default configuration, which you can edit.



### To create an AppServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **AppServer**. The **AppServer Configuration** page appears.
4. Type the name of the new AppServer in the field provided.

---

**Note:** The AppServer name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured AppServer names.

---

5. Click **Save**. The **AppServer Configuration** page appears, allowing you to configure the AppServer's properties.

Each new AppServer instance you create uses the default configuration. However, the port number must be unique for each AppServer instance in order for the instance to operate properly. Once you create a new instance, you are prompted to change the port number.

## Starting and stopping an AppServer

From the management console, you can start either a local or a remote AppServer. You must first connect to the AdminServer that manages the AppServer you want to start.

You can also choose to start an AppServer automatically when the AdminServer starts.

---

**Note:** Prior to starting an AppServer using SonicMQ ServerConnect, you must first configure the AppServer as SonicMQ ServerConnect enabled.

---



### To start an AppServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer instance you want to start. The **Details** page for that instance appears in the detail frame.
3. Click **Broker Control**. The instance's **Control** page appears.
4. Click **Start AppServer**.



## Starting an AppServer automatically

You can set an AppServer to start automatically when the AdminServer starts.



### To start an AppServer automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer instance you want to start automatically. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The **AppServer Configuration** page appears.
4. Click **Edit**.
5. Select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.

## Stopping an AppServer

From the management console, you can directly stop a local AppServer on the machine where it runs.

---

**Note:** To stop a remote AppServer, access the corresponding local AppServer instance selected in the container of the host where it runs.

---



### To stop an AppServer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer instance you want to stop. The **Details** page for that instance appears in the detail frame.
3. Click **Broker Control**. The **AppServer Control** page appears.
4. Click **Stop AppServer**.

## Adding or trimming AppServer agents

At run time, you can increase the number of AppServer agents running and available to service AppServer client requests by having the AppServer add a specified number of AppServer agents to its process pool.

You can also trim running AppServer agents, down to the Minimum Servers property setting, from the management console.

### More information about trimming AppServer agents

To better understand the use of the Agent Auto-trim feature, consider a scenario involving a broker serving a client/server application running in stateless mode. Assume that the following property values are set:

- Minimum Agent/Server Instances: 10
- Initial Agent/Server Instances to Start: 10
- Maximum Agent/Server Instances: 50
- Agent/Server Auto-trim Interval: 1800

When the broker starts, it starts 10 agents. This also marks the start of the first Agent/Server Auto-trim Interval. Assume that the broker is idle for 30 minutes (it is just before people arrive for work, for instance). At the end of the 30-minute interval, the highest number of agents that were busy at the same time would be zero. (Since there were no requests made of any of the agents, none of the agents was busy.) However, since zero is less than the Minimum Agent/Servers Instances value of 10, no automatic trimming takes place. This marks the start of the second interval.

As the second interval starts, employees begin their workday. There is a sudden demand for agents; in fact, the broker receives so many requests that it must start 20 additional agents. Assume that the high volume of workload, and demand on the agents, continues through the second interval. At the interval's end, the maximum number of busy agents is 30, which is the number the broker scaled up to in response to high demand. Thirty agents are running, and the maximum busy level is 30. No trimming occurs.

During the next interval, the request load diminishes somewhat. At the end of the interval, only 20 agents are concurrently busy. The maximum number of busy agents for the interval remains at 30, since 30 were busy at the start of the interval. Thirty agents are running, and the maximum busy level is 30. No trimming occurs.

Assume that all employees have to attend a company-wide meeting that begins midway through the interval. The highest number of agents busy at that same time was 20. At the end of the interval, there are 30 agents running, with zero agents currently busy. Since the maximum busy level is 20 (at the start of the interval), and the number of running agents is 30 (all are available since everyone is at the company meeting), the broker attempts to trim back the number of agents to 20.

Assume that the company meeting ran for another half-hour, and the entire next interval elapsed without any requests. The maximum busy agents would be zero, with 20 currently running. Since the Minimum Agent/Server Instances property value is set to 10, the broker attempts to trim back to 10 agents.

Keep in mind that the lower you set the Agent/Server Auto-trim Interval value, the more quickly the broker reacts to a drop in the workload. Under a reasonably steady work load, the broker should trim the number of running agents to the number needed. However, if the workload fluctuates a great deal and the Agent/Server Auto-trim Interval is set too low, the broker may trim agents that it must then restart. As stated earlier, repetitive trimming and restarting causes the system to waste resources on unnecessary process management.



### To add or trim AppServer agents:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer broker for which you want to add or trim agents. The **Details** page for that broker appears in the detail frame.
3. If the AppServer is not started, click **Broker Control**. Then click **Start AppServer**.
4. When the AppServer is running, return to the **Details** page.
5. Click **Server Pool Control**.
6. Review the **Server pool initial configuration**, the **Servers state**, and the **Server Pool summary** information. (You can view detailed information about an agent by clicking on its PID.)
7. To add or trim servers, choose the corresponding option in the **Add/Trim** dropdown menu, and type the number of servers in the box provided.
8. Click **Submit**. A confirmation message appears.

## Viewing the status of an AppServer

You can view the status of any running AppServer instance through the management console. Status and/or statistical details appear in the following different views:

- From the AppServer instance's **Details** page.
- From the **Broker Control** link on the AppServer instance's **Details** page.
- From the **Broker PID** link within the instance's **AppServer Control** page.
- From the **Server Pool Control** link on the AppServer instance's **Details** page.

You can also add and trim AppServer agents from the **Server Pool Control** page.



### To view the status of an AppServer instance from the Details page:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired AppServer instance. The **Details** page for that instance appears in the detail frame and displays the following information:
  - **Host** — The name of the host machine
  - **Broker** — Whether the broker is active or not running
  - **Operating mode** — One of four possible modes can be reported: **Stateless**, **State-reset**, **State-aware**, and **State-free**
  - **Broker statistics available** — Whether broker statistics are being collected

- **Servers available** — 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**
- **Should register with NameServer** — The status of **True** or **False** to indicate whether or not the broker resource is registered with a NameServer



### To view the status of an AppServer instance from the Broker Control link:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired AppServer instance. The **Details** page for that instance appears in the detail frame.
3. Click **Broker Control**. The **AppServer Control** page for that instance appears and displays the following information:

- **Broker name** — The name of the selected AppServer broker
- **Host** — The name of the host machine
- **Port** — The port number
- **Broker PID** — The broker's process identification number (PID).
- **Status** — The running status of the broker. Possible values are:
  - **ACTIVE** (the broker is currently running)
  - **Not Running** (the broker is not currently running)

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.

- **Operating mode** — One of four possible modes can be reported: **Stateless**, **State-free**, **State-aware**, or **State-reset**
- **Broker statistics available** — Whether collected broker statistics are available
- **Enabled** — Whether the broker is enabled
- **Collect Statistics** — Whether statistics are being collected



### To view the status of an AppServer instance from the Server Pool Control link:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired AppServer instance. The **Details** page for that instance appears in the detail frame.

3. Click **Server Pool Control**. The **Server Pool Control** page for that instance appears and displays the following information:

- **Initial number of servers to start** — The value referenced when the AppServer broker starts AppServers.
- **Minimum servers** — The minimum number of AppServers that must be simultaneously running before the AppServer broker starts 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 automatically starts the additional servers needed to maintain the minimum.

If you set a trim value that requires trimming the number of servers to below the number specified for this field, a message appears.

- **Maximum servers** — The maximum number of AppServer processes that can be running simultaneously. Add requests you initiate that will exceed the specified maximum are not fulfilled.
- **Licensed servers** — The total number of licensed servers.
- **Active servers** — The number of AppServers currently running.
- **Busy servers** — The number of AppServers handling ABL client requests
- **Locked servers** — The number of AppServers servicing a bound connection. (This state applies to a stateless AppServer.)
- **Available servers** — The number of AppServers available to handle broker requests.

A server pool summary provides details about the following:

- **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, to kill the process. See the section about killing an AppServer process in *OpenEdge Management: Servers, DataServers, Messengers, and Adapters* for more information.
- **State** — The current execution state of the agent.
- **Port** — The TCP/IP port number used by the agent.
- **nRq** — The number of messages sent to the agent.
- **nRcvd** — The number of messages received by the agent.
- **nSent** — The number of requests sent by the agent.
- **CPU Use** — The percentage of CPU user and system time consumed by a process.
- **Memory Use** — The amount of virtual memory (in Kbytes) consumed by a process.
- **Started** — The time stamp indicating when the agent started
- **Last Change** — The time stamp indicating when the agent last changed execution state

## Deleting an AppServer

You can delete any AppServer instance listed in the management console as long as the instance is not running. (The **Delete** button is not available if the instance is running.) If the instance is running, first stop it and then delete it.



### To delete an AppServer instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer instance you want to delete. The **Details** page for that instance appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Viewing the AppServer log files

The AppServer log file viewer allows you to examine the contents of a log file through an HTML interface.

For details, see the section about using the AppServer log file viewer in *OpenEdge Management: Servers, DataServers, Messengers, and Adapters*.

## For additional AppServer information

You can find additional information about using OpenEdge Management with the AppServer in *OpenEdge Management: Servers, DataServers, Messengers, and Adapters*. See the document for details, including specifics about the following:

- Working with the AppServer log file monitor and viewer
- Killing an AppServer process
- Listing AppServer Client connections

## AppServer Internet Adapter configuration and administration

With the AppServer Internet Adapter (AIA) you can make AppServer or SonicMQ Adapter application services available over the Web to ABL applications, and you can make AppServer application services available over the Web to .NET and Java Open Clients. The AIA is a Java Servlet that is invoked by a Java Servlet Engine (JSE).

When you use the URL syntax for one of the following connection protocols to connect standard ABL clients (including AppServer and WebSpeed agents, and WebClients) or .NET and Java Open Clients to an AppServer, or to connect standard ABL clients to a SonicMQ Adapter:

- **HTTP** — Directs the client to connect through an AIA instance using an unsecured Internet connection to the Web server that hosts the AIA
- **HTTPS** — Directs the client to connect through an AIA instance using a Secure Sockets Layer (SSL) Internet connection to the Web server that hosts the AIA

---

**Note:** You can also use the AppServer protocol to allow a client to connect directly to an AppServer or a SonicMQ Adapter using the TCP/IP protocol with or without SSL tunneling. This is the preferred option for client access over an intranet.

However, if you want to use a secure Internet connection to the Web server and also secure the connection between the AIA and the AppServer (or SonicMQ Adapter), you can configure the AIA to use SSL for its own connection to the AppServer (or SonicMQ Adapter).

---

### Working with the AppServer Internet Adapter

To work with an AIA instance, double-click the **AppServer Internet Adapter** folder from the management console's list pane. There is one pre-defined default AppServer Internet Adapter (Aia1).

You can perform the following AppServer Internet Adapter actions:

- [Configuring an AppServer Internet Adapter](#)
- [Editing an AppServer Internet Adapter configuration](#)
- [Creating an AppServer Internet Adapter](#)
- [Deleting an AppServer Internet Adapter broker](#)
- [Viewing the AppServer Internet Adapter log file](#)



## Configuring an AppServer Internet Adapter

You can view or modify the following configuration properties of an AppServer Internet Adapter broker from the management console:

- General
- Controlling NameServer
- Logging Setting
- SSL
- Advanced Features

You can also create a new AppServer Internet Adapter instance, delete an instance you no longer need, or view its log file.

Each of these tasks is described in this chapter.



### To view or modify AppServer Internet Adapter configuration properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **AppServer Internet Adapter** category, and select an AppServer Internet Adapter instance. The **Details** page for that instance appears in the Detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **AppServer Internet Adapter Configuration** page opens with the **General** properties in focus.

From this page, you can do the following:

- View a read-only display of the AppServer Internet Adapter instance's properties, as described in [Table 5–15](#) through [Table 5–19](#). All of these tables follow this procedure.
- Click **Edit** to modify the properties. For details, see the “[Editing an AppServer Internet Adapter configuration](#)” section on page 5–37.

**Table 5–15: AppServer Internet Adapter General properties**

Property	Description
<b>Idle connection timeout</b>	<p>Timeout value, in seconds, between an AppServer client and an AIA instance. If a connection is idle for more than the specified timeout value, then the AIA instance automatically disconnects from the AppServer.</p> <p>The default is 3600.</p>
<b>Secure port</b>	<p>Identifies the port on which the Web server listens for requests that specify the HTTPS protocol. This value is used only when the <b>HTTPS enabled</b> property is set to on. If you do not specify a value for this property, and the <b>HTTPS enabled</b> property is set to on, then the default value for <b>Secure Port</b> is 443.</p>
<b>HTTPS enabled</b>	<p>Determines how an AppServer connection using the AIA connects to the AIA's Web server host. If you select this option, clients must use HTTPS to connect and send encrypted data across the Internet to the Web server. If you do not select this option, clients can use HTTP to connect and send data across the Internet without encryption.</p> <p>When enabled, if an AppServer client uses HTTP to connect to an AppServer, the AIA returns an HTTPS redirection URL to the client. The client automatically retries the connection using the HTTPS URL.</p> <p>The default setting for this check box is enabled.</p>

**Table 5–16: AppServer Internet Adapter Controlling NameServer properties**  
(1 of 2)

Property	Description
<b>Connect using NameServer</b>	A checkbox that indicates whether to connect to an AppServer using a Controlling NameServer.
<b>Controlling NameServer</b>	Identifies the NameServer that the AIA uses to find a broker that supports the correct application services.

**Table 5–16: AppServer Internet Adapter Controlling NameServer properties**  
(2 of 2)

Property	Description
<b>Minimum NameServer client port</b> and <b>Maximum NameServer client port</b>	<p>These properties are used to limit the range of ports of an AIA communicating with the NameServer via UDP. This is important when the AIA and NameServer are separated by a firewall.</p> <p>The value for these two properties must be a number between 1024 and 65535 inclusive (or 0), and <b>Minimum NameServer client port</b> must be less than or equal to <b>Maximum NameServer client port</b>.</p> <p>If the minimum and maximum values are both set to zero, then a random port number within the implicit fixed range will be dynamically assigned.</p> <p>If both minimum and maximum values are set to the same port number, then this UDP port number will be used when communicating with the NameServer.</p> <p>The default value for both minimum and maximum is 0.</p>
<b>AppServer host name</b>	<p>If you did not select the <b>Connect using NameServer</b> check box and the AIA will be directly connected to an AppServer or SonicMQ Adapter, you must specify the AppServer's or SonicMQ Adapter's host location.</p>
<b>AppServer port</b>	<p>If you did not select the <b>Connect using NameServer</b> check box and the AIA will be directly connected to an AppServer or SonicMQ Adapter, you must specify the AppServer's or SonicMQ Adapter's port.</p>

**Table 5–17: AppServer Internet Adapter Logging Setting properties** (1 of 3)

Property	Description
<b>Log filename</b>	<p>The AIA log filename. Enter a valid path name.</p> <p>The installed default is @{WorkPath}\Aia1\aia.log</p>
<b>Logging level</b>	<p>A value that specifies the amount of information to be written to the AIA log. Select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul> <p>The default is <b>Basic</b>.</p>
<b>Append to log file</b>	<p>A check box that indicates whether or not a new server log file should be created when the AIA is started.</p> <p>To create a new AIA log file each time the AIA is started, even if the AIA log file already exists, clear the <b>Append to log file</b> check box.</p> <p>To append log entries to the existing AIA log file, select the <b>Append to log file</b> check box.</p> <p>The <b>Append to log file</b> check box is selected by default.</p>

**Table 5–17: AppServer Internet Adapter Logging Setting properties** (2 of 3)

Property	Description
<b>Logging entry types</b>	<p>The AIA log entry type turns on logging for different AIA actions based on the logging level. The AIA log entry types provide the following information:</p> <ul style="list-style-type: none"> <li>• <b>AiaDefault</b> — Combines AiaMgmt and AiaProp. This is the default entry type for the AIA component.</li> <li>• <b>AiaMgmt</b> — Logs messages about regular AIA operations, such as AIA version information, instantiation, informational details, and information on the AIA connection pool according to the following logging level settings: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs instantiation and informational messages</li> <li>– <b>Verbose</b> — Logs the connection pool information</li> <li>– <b>Extended</b> — Logs the same information as <b>Verbose</b></li> </ul> </li> <li>• <b>AiaProp</b> — Logs the properties set for the specific AIA instance. These properties are set in the <code>ubroker.properties</code> file. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>AiaRqst</b> — Logs information on the GET and POST commands sent by the client and the response back to the client. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>AiaUBroker</b> — Logs information on the communication between the AIA and the Unified Broker (AppServer or SonicMQ Adapter) according to the following Logging level settings: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Provides most of the information</li> <li>– <b>Verbose</b> — Logs a binary dump of messages exchanged between the client and the broker/server process</li> </ul> </li> </ul>

**Table 5–17: AppServer Internet Adapter Logging Setting properties** (3 of 3)

Property	Description
<b>Log file threshold size</b>	A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.
<b>Maximum number of log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of AIA log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p style="text-align: center;">filename.#####.extension</p>

**Table 5–18: AppServer Internet Adapter SSL properties**

Property	Description
<b>Enable SSL client connections</b>	If checked, this property directs that the connection to the AppServer or SonicMQ Adapter use SSL tunneling.
<b>Disable client verification of SSL host name</b>	If checked, this property turns off host verification for an SSL connection. If cleared, the client compares the host name for the AppServer or SonicMQ Adapter specified by AppServer Location Host Name (or as returned by any Controlling NameServer) with the Common Name specified in the server digital certificate, and raises an error if they do not match. With this parameter specified, the client never raises the error.
<b>Disable SSL session reuse</b>	If this property is checked, the connection does not reuse the SSL session ID when reconnecting to the same AppServer or SonicMQ Adapter.

**Table 5–19: AppServer Internet Adapter Advanced Features properties**

Property	Description
<b>Internal administrative command</b>	Provides permission to see whether or not the AIA is running successfully.  The check box is cleared by default.
<b>Authorized IP list</b>	If the <b>Internal administrative command</b> check box is selected, this box shows a comma-separated list of IP addresses that are allowed to access the AIA Adapter's internal administrative commands.  To add or delete an IP address, click <b>Edit</b> and type or remove an IP address in the box.  To allow any IP address to access the AIA Adapter's internal commands, leave the box blank. This is the default.

## Editing an AppServer Internet Adapter configuration

You can edit the configuration of an AppServer Internet Adapter.



### To edit an AppServer Internet Adapter configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer Internet Adapter whose configuration you want to edit. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The **Configuration** page appears.
4. Click **Edit**. (For details about the properties, see [Table 5–15](#) through [Table 5–19](#).)
5. Make the configuration changes, and then click **Save**.

## Creating an AppServer Internet Adapter

You can create an AppServer Internet Adapter instance.



### To create an AppServer Internet Adapter instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **AppServer Internet Adapter**. The **AppServer Internet Adapter Configuration** page appears.

4. Type the name of the new instance in the field provided.

---

**Note:** The AppServer Internet Adapter name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured AppServer Internet Adapter names.

---

5. Click **Save**. The **AppServer Internet Adapter Configuration** page appears, showing the configuration details and allowing you to configure the instance's properties.

## Viewing the AppServer Internet Adapter log file

You can view the log file for an AppServer Internet Adapter log file by using the log file viewer. The log file viewer allows you to examine the log file through an HTML interface.



### To access the log file viewer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired instance. The **Details** page for that instance appears in the detail frame.
3. Click **Log File Viewer**.

You can work with the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display 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 at 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, regardless of 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 **Previous** 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**.
- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.
- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## Enabling or disabling an AppServer Internet Adapter broker

You can enable or disable an AppServer Internet Adapter broker.



### To enable or disable an AppServer Internet Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer Internet Adapter broker you want to enable or disable. The **Details** page for that broker appears in the detail frame.
3. Click **Control**. The broker's **Control** page appears.
4. Click **Edit**, and either select the **Enabled** check box to enable the broker or clear the check box to disable the broker.
5. Click **Save**.

## Deleting an AppServer Internet Adapter broker

You can delete an AppServer Internet Adapter (AIA) instance using the management console. The instance is also deleted from the `ubroker.properties` file.

If you delete an AIA instance that is running, the instance and all associated properties are deleted from the `ubroker.properties` file. However, the AIA instance continues to run until you use the Java servlet engine to shut down the AIA.



### To delete an AppServer Internet Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the AppServer Internet Adapter broker you want to delete. The **Details** page for that broker appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Configuring OpenEdge Web Services

---

You can work with Web Services Adapter instances in OpenEdge Management and OpenEdge Explorer, as described in the following sections:

- [Configuring and managing Web services](#)
- [Managing Web Services Adapter instances](#)
- [Working with Web services](#)

## Configuring and managing Web services

The Web Services Adapter (WSA) is a Java servlet that exposes OpenEdge AppServer applications as Web services. The WSA is installed and runs in the context of a Java servlet engine (JSE) that runs in the context of a Web server.

To expose AppServer applications as Web services, the WSA serves a dual role:

- As a gateway between the SOAP request messages, which Web services and Web service clients exchange, and ABL applications on the AppServer, which execute Web service requests
- As an application server that hosts, manages, and provides communications and run-time support for multiple deployed Web service applications

### SOAP-AVM gateway

For any given Web service, the WSA is a gateway between SOAP and the ABL Virtual Machine (AVM) running on the AppServer. As such, the WSA receives Web service method calls and parameters sent as SOAP messages from Web service clients and converts them into ABL function and procedure calls that the WSA invokes on a specified AppServer. In response, the WSA also receives the return and output parameter values sent from the called ABL functions and procedures on the AppServer and converts them into SOAP messages that it sends back to the calling Web service clients.

### Web Service Application Server

The WSA is where you deploy a Web service that you want to make available to clients. Once deployed, it also allows you to manage the Web service individually and as part of a group of Web services. This includes such options as making Web service definitions available to potential clients as downloadable Web Services Description Language (WSDL) files and making one or more Web services available as applications for access by clients.

To initially create Web service definitions, you use ProxyGen to define an existing set of AppServer procedures and functions in terms of Web service objects. ProxyGen generates a Web Service Mapping (WSM) file. This WSM file stores the Web service object definitions in a form similar to a WSDL file, but without the final deployment information targeted to a specific WSA installation.

Thus, when you deploy a Web service to a WSA, you specify the information including the WSM file into a WSDL file that a specific WSA can use to provide the Web service definition to Web service clients. This same process also creates a Web Service Application Descriptor (WSAD) file that provides the Web service definition to a specific WSA in a form that allows it to host the deployed Web service and manage its run-time behavior.

## Managing Web services with the WSA

The WSA is installed in a JSE as a single Web application that you can instantiate as one or more servlets (WSA instances). Each WSA instance provides a single point of management for one or more deployed Web services. Thus, to begin working with the WSA in OpenEdge Management or OpenEdge Explorer:

1. Create a new WSA instance if none exists or you need another, and configure it for operation. (The WSA comes installed with one local WSA instance, `wsa1`.)
2. For a WSA instance where you want to deploy a new Web service, optionally review and update the default property settings that govern the run-time behavior of new Web service deployments. You can also adjust these settings for each Web service individually after you deploy it.
3. If you want to import an existing Web service, first export the Web service from the context of the WSA instance where it is currently hosted.
4. Deploy the new Web service or import the existing Web service into the context of a WSA instance that you have configured.
5. As you client-test the Web service, review the WSA instance status and statistics, and update any WSA instance properties as needed. Such changes affect all Web services deployed to the same WSA instance.

---

**Note:** To apply changes to most WSA properties, you must restart the JSE where the WSA is installed. You can make temporary changes to selected WSA properties at run time without restarting the JSE. However, you must update the WSA instance configuration to make these changes permanent.

---

6. As you client-test the Web service, review the Web service status and statistics, and update any Web service properties as needed. Such changes affect only the selected Web service. For most of these changes, you must disable the Web service before you make them, then enable the Web service afterward to evaluate the effect.

## Web service session models and the WSA

Web service applications in OpenEdge are designed to run in one of two AppServer session models:

- Session-Managed
- Session-Free

The session model of a Web service application significantly affects how the WSA manages the Web service at run time. In any case, the session model specified for a Web service (when defined using ProxyGen) must match the session model of the AppServer to which it is bound. Otherwise, the WSA returns an error when a client attempts to access the Web service.

### Session-Managed Web services

The WSA handles session-managed Web services in exactly the same way that other OpenEdge clients access an AppServer running in state-aware, state-reset, or stateless operating mode. That is, session-managed Web services are single-threaded. Thus, the WSA maintains a single network session to connect the Web service and all its objects to a single AppServer on behalf of a single Web service client. All requests from the client to the same Web service objects use the same AppServer connection. This session model provides full support on the AppServer side for context management, and makes available the capabilities of the AppServer to support transaction-oriented interactions across multiple client requests.

### Session-Free Web services

The WSA handles session-free Web services in exactly the same way that other OpenEdge clients access a traditional AppServer running in state-free operating mode. For session-free Web services, the WSA maintains a pool of AppServer TCP/IP connections (connection pool) for use by all clients of a single Web service. This allows multiple clients to access the same Web service concurrently; that is, session-free Web services are multi-threaded. The WSA can do this because a session-free AppServer maintains no client connections, but accepts and responds to client requests one at a time regardless of origin and with no context maintained from one request to another. This session model thus supports the industry standard for Web service interactions that focuses on small, atomic, and complete transactions in a single client request.

The connection pool maintains a list of available connections to session-free AppServers that support the same application service on behalf of all clients using the corresponding Web service. The WSA can then service as many client requests at one time as AppServer connections in the connection pool. It also queues any additional client requests for the first available AppServer that completes a client request.

---

**Note:** Session-free Web services can support Web service objects that require dedicated AppServer connections. However, typical session-free Web services avoid the use of these types of Web service objects because they undercut the benefits of connection-less interactions.

---

## Managing Web Services Adapter instances

You can manage Web Services Adapters by:

- [Creating a new Web Services Adapter instance](#)
- [Configuring a Web Services Adapter instance](#)
- [Editing a Web Services Adapter instance configuration](#)
- [Starting Web Services Adapter instances](#)
- [Setting Web service defaults](#)
- [Deploying a Web service to a Web Services Adapter instance](#)
- [Importing a Web service to a Web Services Adapter instance](#)
- [Listing the deployed Web services for a Web Services Adapter instance](#)
- [Viewing the status of a Web Services Adapter instance](#)
- [Viewing Web service statistics](#)
- [Changing Web Services Adapter Instance run-time properties](#)
- [Stopping Web Services Adapter instances](#)
- [Deleting a Web Services Adapter instance](#)

Each of these tasks is described in the following sections.

### Creating a new Web Services Adapter instance

OpenEdge Management and OpenEdge Explorer provide a sample Web Services Adapter (WSA) instance (wsa1). You can edit the wsa1 sample configuration, and you can create new WSA instances.

---

**Note:** To complete the creation of a WSA instance after creating it in the management console and before you can deploy Web services to it, you must update your JSE configuration and edit the WSA's `web.xml` file. Add a servlet and servlet-mapping element for the instance, then restart your JSE. This creates the root sub-path for the new instance, initializes and runs it.

For more information, see *OpenEdge Development: Web Services* (in the Product Documentation section of the Progress Software Developer's Network Web site) and your JSE documentation.

---

**To create a new WSA instance:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **WebServices Adapter**. The **WebServices Adapter Configuration** page appears.
4. Type the name of the new Web Services Adapter in the **New Web Services name** field.

---

**Note:** The Web Services Adapter name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured NameServer names.

---

5. Type the URL in the **Url** field.

This is a URL that directly addresses the WSA instance on the Internet and serves as the root for other URLs required to access Web services and other functions of the WSA.

Form the root URL according to the following format:

`http://host[:port][/jse-context]/webapp-context/servlet-instance`

Where:

- **host** — The name of the machine where the WSA is installed (Default: localhost).
- **port** — The HTTP port number on the WSA machine (Default: 80).
- **jse-context** — The JSE context path name is present only when you have a Web server between your Java Servlet Engine (JSE) and the client application. It specifies the path name that you define in your Web server configuration to send client requests to your JSE.
- **webapp-context** — The Web application context path name always exists and specifies the path name that you define in your JSE configuration to identify the WSA as a servlet application.
- **servlet-instance** — The servlet-mapping name for the WSA instance (WSA Web application) specified in *OpenEdge-install-dir\servlets\wsa\WEB-INF\web.xml*. This is also the name by which the WSA instance appears in the Web Services Adapter folder in the management console.



6. Select the WSA installation location for this instance from these choices:
  - **local** — The WSA is installed on the machine where the selected AdminServer is running.
  - **remote** — The WSA is installed on a network machine that is remote from the selected AdminServer.
7. Click **Save**. The new WSA instance appears under the **Web Services Adapter** folder.

You can now configure the properties of the new instance. Note that when you create a new instance, it has a copy of the default configuration. Initially, the instance is enabled only for administration, allowing you to modify its properties and deploy Web services to it without allowing premature user access to an incomplete configuration.

## Configuring a Web Services Adapter instance

Before deploying Web services, you must configure a Web Services Adapter instance to host the Web services that you want to deploy or import. To configure a WSA instance, you set properties for it similar to any other Unified Broker product.

After the WSA is configured, you can import or deploy a Web service and configure additional properties that affect execution of the Web service. The Web service properties that are available for configuration depend on the session model of the Web service (session managed or session free). To verify the session model of a Web service, check its status. For details on checking Web service status, see the [“Viewing the status of a Web service”](#) section on page 6–26. For details on configuring Web service properties, see the [“Updating a deployed Web service”](#) section on page 6–27. For details on the properties available on a Web service, see [Appendix A, “Reference to OpenEdge Web Service Properties.”](#)

You must disable a Web service before you can configure most properties on it. You can verify if the Web service is disabled by checking its status. The only properties that you can configure for an enabled Web service include:

- serviceFaultLevel
- serviceLoggingLevel

If you set other properties while the Web service is enabled, the property value changes take effect only after you disable and then enable the Web service again. For details about enabling or disabling a Web service, see the [“Enabling or disabling a Web service for client access”](#) section on page 6–25.

To configure a Web Services Adapter instance, you can view or edit properties in the following categories:

- Location
- Proxy Server Setting
- WSDL
- Logging Setting
- Security
- Advanced Features

You can also create a new Web Services Adapter instance, view its status, or delete an instance you no longer need.

Each of these tasks is described in the following sections.

### Setting Web Services Adapter instance properties

You can set location, proxy server setting, WSDL, logging setting, security, and advanced features properties for a Web Services Adapter instance.



#### To set Web Services Adapter instance properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **Web Services Adapter** category, and select an instance. The **Details** page for that instance appears in the detail frame.
4. If your Web server requires that you log in, click **Login** in the **Command and control** section of the page. Type your user name and your password, and then click **Submit**.
5. Click **Configuration**.

From this page, you can do the following:

- View a read-only display of the Web Services Adapter properties as described in [Table 6-1](#) through [Table 6-6](#). All of these tables follow this procedure.
- Click **Edit** to modify the properties. For details, see the [“Editing a Web Services Adapter instance configuration”](#) section on page 6-16.

### Location properties

You set the properties, shown in [Table 6-1](#), in this category when you first create the WSA instance. For more information see the [“Creating a new Web Services Adapter instance”](#) section on page 6-5.

You cannot change the location properties. Instead, you must delete and then recreate the WSA instance with the new values. For more information, see the [“Deleting a Web Services Adapter instance”](#) section on page 6-24.

**Table 6–1: Web Services Adapter Location properties**

Property	Description
<b>Location</b>	<p>A read-only property that indicates where the WSA is installed:</p> <ul style="list-style-type: none"> <li>• <b>local</b> — The WSA is installed on the machine where the selected AdminServer is running. You can configure all the properties of a local WSA.</li> <li>• <b>remote</b> — The WSA is installed on a network machine that is remote from the selected AdminServer.</li> </ul>
<b>URL</b>	A read-only property set to the base URL for the WSA.
<b>Web Services Protocol Value</b>	<p>The protocol for all Web service URLs inserted into WSDL documents that are generated for this WSA instance during Web service deployment. The host, port, and other path information specified for the WSA URL when the instance was originally created remains unchanged.</p> <p>You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>Duplicate WSA Url value</b> — Uses the original protocol specified for the WSA URL</li> <li>• <b>Use HTTP protocol</b> — Replaces the original WSA URL protocol with HTTP</li> <li>• <b>Use HTTPS protocol</b> — Replaces the original WSA URL protocol with HTTPS</li> </ul> <p>Installed Default: Duplicate WSA Url value</p>

**Proxy Server Setting properties**

The properties in this category, as shown in [Table 6–2](#), apply to both local and remote WSA instances. They specify the information required to access any proxy server for the network on which the WSA resides.

**Table 6–2: Web Services Adapter Proxy Server Setting properties** (1 of 2)

Property	Description
<b>Proxy host</b>	<p>The TCP host name for the proxy server</p> <p>Installed Default: None</p>
<b>Proxy port</b>	<p>The TCP port number of the proxy service</p> <p>Installed Default: 0</p>
<b>Proxy username</b>	<p>A valid username required to access the proxy server</p> <p>Installed Default: None</p>

**Table 6–2: Web Services Adapter Proxy Server Setting properties** (2 of 2)

Property	Description
<b>Proxy password</b>	The password for the specified username  Installed Default: None
<b>Proxy password Confirm</b>	The password retyped for confirmation

**WSDL**

The properties in this category, as shown in [Table 6–3](#), control access to WSDL files and information.

**Table 6–3: Web Services Adapter WSDL properties**

Property	Description
<b>Enable WSDL retrieval</b>	If activated, this check box setting allows network users to access WSDL files deployed to this WSA instance. Otherwise, users cannot access any WSDL files deployed to this WSA instance.  Installed Default: Activated
<b>WSDL inaccessible page</b>	The relative path and filename of the Web page that you want the WSA instance to return when users attempt to access WSDL files deployed to this instance. The <b>Enable WSDL Retrieval</b> check box is cleared to prevent that access. The path name is relative to your web application (WSA) servlet root directory.  Installed Default: None
<b>Enable WSDL listing retrieval</b>	If activated, this check box setting allows network users to access a Web page that lists the WSDL files deployed to this WSA instance. Otherwise, users cannot access the list of WSDL files deployed to this WSA instance.  Installed Default: Activated
<b>WSDL listing page</b>	The path and filename of the Web page that you want the WSA instance to return when users request a listing of the WSDL files deployed to this instance. The WSA returns this page only when the <b>Enable WSDL listing retrieval</b> check box is activated. The path name is relative to your web application (WSA) servlet root directory.  Installed Default: WSDLListing.html

## Logging Setting

The properties in this category, as shown in [Table 6–4](#), control the location and content of the WSA log file.

**Table 6–4: Web Services Adapter Logging Setting properties** (1 of 2)

Property	Description
<b>Working directory</b>	The directory where the WSA writes temporary files when required. Enter a valid path name. Installed Default: @{WorkPath}
<b>Log filename</b>	The WSA log filename. Enter a valid path name. Installed Default: @{WorkPath}\wsa1.wsa.log
<b>Logging level</b>	<p>A value that specifies the amount of information that is written to the WSA log. Select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> - Logging Entry Types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> —Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> —Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul> <p><b>Note:</b> A value of <b>Other</b> in this field indicates an option manually specified in the <code>ubroker.properties</code> file that provides additional diagnostic logging features. You can reset this by choosing from the values in the drop-down list.</p> <p>Installed Default: <b>Basic</b></p>
<b>Append to log file</b>	<p>A check box that indicates whether or not a new server log file should be created when the WSA is started.</p> <p>To create a new WSA log file each time the WSA is started, even if the WSA log file already exists, clear the <b>Append to log file</b> check box.</p> <p>To append log entries to the existing WSA log file, select the <b>Append to log file</b> check box.</p> <p>Installed Default: Selected</p>

**Table 6–4: Web Services Adapter Logging Setting properties**

(2 of 2)

Property	Description
<b>Log message threshold size</b>	<p>An integer value that limits the amount of SOAP message detail written to the log file for clients specified using the <b>Debug Clients</b> property in the <b>Advanced Features</b> category. A value of <b>-1</b> indicates that all SOAP message detail is written to the file. A positive value indicates the number of characters written to the file for each logged SOAP message. Practical values range from 100 to 10000, but any positive value is valid.</p> <p>Installed Default: -1</p>
<b>Logging entry types</b>	<p>A comma-separated list of valid entry types for WSA logging. The only standard entry is WSADefault. Set additional entry types only if instructed to do so by Progress Software Corporation Technical Support.</p> <p>You can specify a separate logging level for each log entry type, and you can use a wildcard (*) to specify multiple entry types. For example:</p> <p><code>UBroker.Basic:3,UBroker.ThreadPool:4,UBroker.Client*</code></p>
<b>Log file threshold size</b>	<p>A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.</p>
<b>Maximum number of log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of WSA log files to keep. The specified number represents the maximum number of log files to keep on disk at any time, including the current log file being written to.</p> <p>The default value is 3.</p> <p>When the file becomes equal to or greater than the <b>Log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p><code>filename.#####.extension</code></p>

## Security

The properties in this category, as shown in [Table 6–5](#), control options for authorizing and authenticating access to WSA functions and Web services.

**Table 6–5: Web Services Adapter Security properties**

Property	Description
<b>Enable WSA administration</b>	<p>A check box that indicates whether or not administration functions are available for the selected WSA instance.</p> <p>Installed Default: Selected</p>
<b>WSA administration authentication</b>	<p>A drop-down list that indicates the criteria used to authenticate users for access to WSA administrative functions. The choices include:</p> <ul style="list-style-type: none"> <li>• <b>No user authentication</b> — Any user can perform WSA administration functions.</li> <li>• <b>Use Admin Server user authentication</b> — The AdminServer forwards the same user credentials supplied at connection time to the Web server (or JSE) where the WSA is running in order to perform administration functions.</li> <li>• <b>Prompt user for Web Server username and password</b> — Only users authorized to manage the Web server (or JSE) where the WSA is running can perform WSA administration functions.</li> </ul>
<b>Require WSA administration authorization</b>	<p>A check box that indicates whether or not each user should be validated against selected Administration Role definitions when the user attempts to perform an administrative function on the WSA instance or on any of its deployed Web services.</p> <p>For details, see the <a href="#">“Viewing or modifying the Admin or Operator role”</a> section on page 6–14.</p>
<b>To enable administration roles, select from the following list</b>	<p>Predefined Roles: PSCAdmin, PSCOper</p> <p>For details, see the <a href="#">“Viewing or modifying the Admin or Operator role”</a> section on page 6–14.</p>
<b>Enable Web Services</b>	<p>A check box that indicates whether or not users can access the Web services deployed to the selected WSA instance.</p> <p>Installed Default: Cleared</p>
<b>Require Web Services authorization</b>	<p>A check box that indicates whether or not users must be authorized to access Web services.</p> <p>Installed Default: Cleared</p>
<b>Require WSDL Retrieval authorization</b>	<p>A check box that indicates whether or not users must be authorized to retrieve WSDL files and information.</p> <p>Installed Default: Cleared</p>

## Viewing or modifying the Admin or Operator role

You can define what level of access is allowed for the Web Services Adapter instance: Administrator or Operator. Under **Admin Role Links**, click either **AdminRole.PSCAdmin** or **AdminRole.PSCOper** to see the current read, write, delete, and execute settings for the following:

- **Application defaults** — Governs which operations the role may perform on the default Web service properties that are used to initialize a newly deployed Web service.
- **Application enablement** — Governs which operations the role may perform on the enable state of Web service applications.
- **Application properties** — Governs which operations the role may perform on a Web service's properties.
- **Application statistics** — Governs which operations the role may perform on a Web service's statistical counters.
- **Servlet properties** — Governs which operations the role may perform on the WSA's run-time changeable properties.
- **Servlet services** — Governs which operations the role may perform on the registry of deployed Web service applications.
- **Servlet statistics** — Governs which operations the role may perform on the WSA's statistical counters.

To edit the settings, click **Edit**. To create a new AdminRole name, click **Create**.

## Advanced Features

The properties in this category, shown in [Table 6–6](#), support less commonly-used or special-purpose features, some of which might relate to the properties of other categories.

**Table 6–6: Web Services Adapter Advanced Features properties** (1 of 2)

Property	Description
<b>WSA administration SOAP action</b>	<p>A string placed in the SOAPAction HTTP header when the AdminServer forwards administration operations to the WSA. The SOAPAction HTTP header is required for all HTTP messages that carry SOAP messages and is used by intervening security servers (such as firewalls) to determine if each HTTP message is allowed to pass through to its destination. The installed default is a Universal Resource Number URN identifying the WSA administration Web service, but you can also specify any string required by intervening security servers on the network.</p> <p>Installed Default: urn:services-progress-com:wsa-admin:01</p>



**Table 6–6: Web Services Adapter Advanced Features properties** (2 of 2)

Property	Description
<b>HTTP error page</b>	<p>The path and filename of a static Web page that you want returned to a Web service user if the request results in an HTTP error, such as a service unavailable or an authorization error. The path name is relative to your web application (WSA) servlet root directory.</p> <p>Installed Default: <code>httperror.html</code></p>
<b>Debug clients</b>	<p>A list of dot-formatted IP addresses for Web service client machines whose HTTP and SOAP messages you want dumped to the WSA log file. The WSA dumps each SOAP message up to the limit specified by the <b>Log message threshold</b> property setting in the <b>Logging Setting</b> category.</p> <p>To add a client machine to the list, enter its dot-formatted IP address in the Client Host field, then choose the <b>Add</b> button to add it to the list. You can also enter multiple, comma-separated addresses in the field at one time to dump messages for multiple clients, or enter a single asterisk (*) to dump the messages for all clients.</p> <p>To remove whole lines of addresses from the list, select one or more lines in the list and click <b>Delete</b>.</p> <p>To edit a single line of addresses, select the line in the list, and after modifying the line click <b>Add</b> to add it to the list. The original line remains. Select the original line and click <b>Delete</b> to remove it from the list and leave the edited line in its place.</p> <p><b>Note:</b> The list automatically adds new lines in increasing order of collation (numerical, for valid IP addresses). If you add an invalid IP address to the list, the WSA never matches it with a message to dump in the log file, and thus effectively ignores it.</p> <p>Installed Default: None</p>

## Editing a Web Services Adapter instance configuration

You can edit the configuration properties of a Web Services Adapter (WSA) instance from the management console.



### To edit Web Services Adapter properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **Web Services Adapter** category, and select a Web Service Adapter instance. The Details page for that instance appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **Web Services Adapter Configuration** page opens.
5. Click **Edit**, and make the changes you want to the properties. (For details about the properties, see [Table 6–1](#) through [Table 6–6](#).)
6. Click **Save**. Any changes you make are also reflected in the `ubroker.properties` file.

You can also make run-time changes in the management console to the values of the following WSA properties:

- **debugClients** — Controls whether to dump HTTP and SOAP protocol message information to the WSA log file, and specifies the clients (if any) whose SOAP messages are to be dumped.
- **enableWsdL** — Controls the WSA's ability to respond to inquiries about WSDL documents for any of the Web service applications it hosts.
- **enableWsdLListings** — Controls the WSA's ability to respond to a request for a list of WSDL documents it has available.
- **loggingLevel** — Specifies the amount of information to be written to the broker log.
- **logMsgThreshold** — Controls the amount of SOAP message information that can be written to the WSA log file.
- **webAppEnabled** — Controls the WSA's ability to accept and process SOAP requests to any of its hosted Web service applications.

These changes last only for the duration of the current Java servlet engine (JSE) session. To make these changes permanent across sessions, you must make the changes in the property settings.

**To edit the WSA instance configuration:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose configuration you want to edit. The **Details** page for the instance appears.
3. Click **Configuration**. The **Web Services Adapter Configuration** page appears.
4. Click **Edit**.
5. Click **Save**.

## Starting Web Services Adapter instances

The Web Services Adapter (WSA) and its instances start up automatically when you start your Web server or Java servlet engine (JSE). For more information, see the documentation for your Web server or JSE.

## Setting Web service defaults

You can view and set the default values used to initialize properties for Web services that you deploy. Any changes to these defaults affect only Web services that you deploy after confirming the change. To set the values of properties for deployed Web services, set the property values individually for each Web service.

---

**Notes:** Unlike WSA instance properties, the defaults for Web service properties are not stored in the `ubroker.properties` file. They are stored in the `default.props` XML file located in the Web application servlet directory for each WSA instance.

---

**To view and set Web service defaults:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where you want to deploy or import a Web service. The **Details** page for that instance appears.
3. Click **Defaults**.
4. Click **Edit**.
5. Select the property whose default value you want to change. For details on the available properties, see [Appendix A, “Reference to OpenEdge Web Service Properties.”](#)
6. Type the new value in the available field. For logical values, typing any value other than **true** results in a setting of **false**.
7. Click **Submit** to change the default value.

To return to the default values, you can click **Reset** from the Defaults page.

## Deploying a Web service to a Web Services Adapter instance

You can deploy a Web service to the context of any available WSA instance using the Web Service Mapping (WSM) file generated using ProxyGen. For more information on generating a WSM file, see *OpenEdge Development: Web Services* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site.

A deployed Web service receives its initial property values from the `default.props` file for the WSA instance. You can either change these default values before you deploy the Web service, or you can change the deployed Web service property values after you deploy the Web service. A Web service always deploys with a disabled status to prevent premature or unintended client access.



### To deploy a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where you want to deploy the Web service. The **Details** page for that instance appears.
3. Click **Deploy**.
4. Type the path and name of the WSM file for the Web service that you want to deploy.
5. Click **Submit**. Verify that the deployment information shown (and which is based on information in the WSM file) is correct. You can modify information in the following fields:
  - **WSM Filename** — The path and name of the WSM file.
  - **Name** — The name used to identify the Web service in the management console and to name the Web service files that are deployed to the WSA instance.
  - **Web Service Namespace** — Any value you choose to uniquely qualify the names for operations and parameters used to define the Web service.
  - **SOAP Action** — A string that the client application is required to place in the SOAPAction HTTP header when accessing a Web service hosted by the WSA instance. The SOAPAction HTTP header is required for all HTTP messages that carry SOAP messages and is used by intervening security servers (such as firewalls) to determine if each HTTP message is allowed to pass through to its destination. The default is a blank string, but it can be any string required by the intervening security servers on the network.

- **Append to SOAP Action** — Indicates whether to append the specified SOAP Action value to any other SOAP Action values generated for the Web service SOAP messages by Web service clients.
  - **WSDL Style/Use** — Specifies the SOAP format to use when sending or receiving SOAP messages for this Web service. The value that you choose is dependent on what your anticipated Web service clients can support and can be any of the following: **RPC/Encoded**, **RPC/Literal**, and **Document/Literal**.
6. Click **Submit** to deploy the Web service with these settings. A confirmation message appears, and the WSM file name is identified in the **Deployed Web Services** section of the WSA instance's **Details** page.

You might need to document additional information about these settings, especially **WSDL Style/Use**, for use by Web service client developers. For more information, see the chapter on client requirements in *OpenEdge Development: Web Services*.

Deployment generates the following files in the WSA instance directory:

- `WebServiceFriendlyName.props` — An XML file containing the current Web service property settings (initially set from `default.props`)
- `WebServiceFriendlyName.wsad` — The Web Service Adapter Descriptor (WSAD) XML file that defines the Web service to the WSA instance
- `WebServiceFriendlyName.wsdl` — The Web Services Description Language (WSDL) XML file that defines the Web service to potential Web service clients

For more information about these files, see *OpenEdge Development: Web Services*.

---

**Caution:** Once you deploy a Web service during development, you can change its definition and deployment information using a Web service update. However, once you deploy and enable a Web service for client access under production conditions, avoid making any changes to this information, as client implementations depend on its stability.

---

After production deployment, to make the same Web service available using different information (for example, to add a new operation or use a different **WSDL Style/Use**), deploy a new Web service with the new information, using a different Web service name in the same WSA context. You could also deploy the Web service to a different WSA context.

However, you can change the run-time properties of a deployed Web service at any time.

## Exporting a Web service

You can export an existing Web service definition to save its deployment information, properties, and other settings. This creates a Web Service Definition (WSD) XML file with a name in the form *filename.wsd*, with a filename that you specify. You can use this file to copy (import) the Web service to another WSA instance or to save it as a backup for the Web service definition.



### To export a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service you want to export is deployed. The **Details** page for that instance appears.
3. Click **Export**.
4. Type the path and file name, ending in `.wsd`, of the file for the Web service you want to export.
5. Click **Submit**. A confirmation message appears.

## Importing a Web service to a Web Services Adapter instance

You can import a Web service to any available WSA instance using the Web Service Description (WSD) file saved during a previous export of the Web service. The import function generates the appropriate Web service properties (`WebServiceFriendlyName.props`), WSAD, and WSDL files to duplicate the original Web service in the context of the selected WSA instance. However, a Web service always imports with a disabled status to prevent premature or unintended client access.



### To import a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where you want to import the Web service. The **Details** page for that instance appears.
3. Click **Import**.
4. Type the path name of the `.wsd` file for the Web service you want to import.
5. Choose **Submit**. Verify that the information shown (and which is based on information in the WSD file) is correct. You can modify information in the following fields:
  - **Name** — The name used to identify the Web service and to name the Web service files that are deployed to the WSA instance.
  - **Web Service Namespace** — Any value you choose to uniquely qualify the names for operations and parameters used to define the Web service.

- **SOAP Action** — A string that the client application is required to place in the SOAPAction HTTP header when accessing a Web service hosted by the WSA instance. The SOAP Action HTTP header is required for all HTTP messages that carry SOAP messages and is used by intervening security servers (such as firewalls) to determine if each HTTP message is allowed to pass through to its destination. The default is a blank string, but can be any string required by the intervening security servers on the network.
  - **Append to SOAP Action** — Indicates whether to append the specified SOAP Action value to any other SOAP Action values generated for the Web service SOAP messages by Web service clients.
  - **WSDL Style/Use** — Specifies the SOAP format to use when sending or receiving SOAP messages for this Web service. Supported formats are **RPC/Encoded**, **RPC/Literal**, and **Document/Literal**.
6. Click **Submit**. A confirmation message appears.

## Listing the deployed Web services for a Web Services Adapter instance

You can list all Web services in the management console that are deployed to a selected WSA instance.



### To list all Web Services deployed to a WSA instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose Web services you want to view. The **Details** page for that instance appears.
3. Click **List**. The list of all Web services deployed to the WSA instance appears.

## Viewing the status of a Web Services Adapter instance

You can view the execution status of a selected WSA instance to see if the instance is running. If the instance is running, the status indicates if administration, Web service application, or WSDL document access is enabled.



### To list the status of a WSA instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose status you want to view. The **Details** page for that instance appears.
3. Under **Operations views**, click **Status**. The following status information appears:
  - Whether the WSA instance is running
  - Whether access to administrative functions, Web service applications (by clients), and WSDL document retrieval is enabled

## Viewing Web Services Adapter instance statistics

You can view (and reset) run-time statistics for a selected WSA instance. Each WSA instance creates and begins accumulating these statistics (Start Time) when it is first started by the Java Servlet Engine (Creation Time). You can also reset the statistics at any point during WSA execution, so that the statistics begin accumulating again from a new Start Time.



### To view the statistics of a WSA instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose statistics you want to view. The **Details** page for that instance appears.
3. Under **Operations views**, click **Statistics**. The following information appears:
  - **Number of SOAP Requests** — Requests for Web service applications
  - **Number of Active Requests** — Requests queued and being acted upon
  - **Number of HTTP Requests** — Total requests received from the HTTP listener, including administrative, WSDL, and Web service requests
  - **Number of WSDL Requests** — Requests for WSDL documents
  - **Number of SOAP Faults** — Error returns from Web service applications



- **Number of Services Disabled** — Web services deployed to this WSA that are disabled from client access
  - **Number of Errors** — Total errors returned by the WSA, with error counts broken out at the bottom of the list for each of several error categories when total errors are greater than zero (0)
4. To reset the statistics and start accumulating all statistics from zero, click **Reset** and confirm that you want to restart the statistics.

## Viewing statistics for deployed Web services

You can view statistical details for deployed Web services.



### To display statistics for deployed Web services:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose run-time statistics you want to see. The **Details** page for that instance appears.
3. Under **Deployed Web Services**, click **Statistics**. Read-only statistics information appears.

You can reset the statistical display by clicking **Reset** on the adapter instance's **Statistics** page.

## Changing Web Services Adapter Instance run-time properties

You can temporarily change some Web Services Adapter (WSA) instance properties at run time without restarting your Java servlet engine (JSE). This is most useful for testing and debugging. The next time you restart your JSE, these settings revert to the current configuration settings for these properties in the `ubroker.properties` file.

To change WSA instance run-time properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance whose run-time statistics you want to see. The **Details** page for that instance appears.
3. Under **Operations views**, click **Run-time Properties**.
4. Click **Edit** to modify any of the following properties:
  - **adminEnabled** — Specifies whether administrator privileges are in effect, either **true** or **false**.
  - **debugClients** — Controls whether to dump HTTP and SOAP protocol message information to the WSA log file, and specifies the clients (if any) whose SOAP messages are to be dumped.

- **enableWsdI** — Controls the WSA's ability to respond to inquiries about WSDL documents for any of the Web service applications it hosts.
- **enableWsdIListings** — Controls the WSA's ability to respond to a request for a list of WSDL documents it has available.
- **loggingLevel** — Specifies the amount of information to be written to the broker log.
- **logMsgThreshold** — Controls the amount of SOAP message information that can be written to the WSA log file.
- **webAppEnabled** — Controls the WSA's ability to accept and process SOAP requests to any of its hosted Web service applications.

5. Click **Submit**.

## Stopping Web Services Adapter instances

The Web Services Adapter (WSA) and its instances stop execution automatically when you shut down your Web server or Java servlet engine (JSE). For more information, see the documentation for your Web server or JSE.

## Deleting a Web Services Adapter instance

You can delete any Web Services Adapter (WSA) instance listed in the management console as long as it has no Web services deployed to it. Undeploy all of its deployed Web services before you attempt to delete the instance.

For more information, see *OpenEdge Development: Web Services* (in the Product Documentation section of the Progress Software Developer's Network Web site) and your JSE documentation.

---

**Note:** To completely remove a WSA instance after deleting it in the management console, you must also remove its corresponding servlet and servlet-mapping elements from the WSA's `web.xml` file and restart your Web server or Java Servlet Engine (JSE).

---



### To delete a Web Services Adapter instance:

1. Stop the Web server.
2. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
3. Expand the Web Services Adapter folder, and select the instance you plan to delete. The **Details** page for that instance appears.
4. Click **Delete**, and then click **OK** to confirm the deletion.

## Working with Web services

You can perform the following Web Services administration tasks from the management console:

- [Enabling or disabling a Web service for client access](#)
- [Viewing the status of a Web service](#)
- [Viewing Web service statistics](#)
- [Updating a deployed Web service](#)
- [Exporting a Web service](#)
- [Undeploying a Web service](#)

### Enabling or disabling a Web service for client access

You can enable any deployed Web service, making it accessible or inaccessible to user requests on the network.



**To enable or disable a Web service for client access:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service you want to enable or disable is deployed. The **Details** page for that instance appears.
3. Choose the deployed application name from the dropdown list, and then click **Select**.
4. Click **Status Enablement**. The list of all Web services deployed to the WSA instance appears.
5. Click **Enable** or **Disable**.
6. Click **OK**. A confirmation message appears.

## Viewing the status of a Web service

You can view status information on a deployed Web service.



### To view the status of a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service whose status you want to view is deployed. The **Details** page for that instance appears.
3. Click **Status Enablement**.

This read-only status presents the following information in a list of label-value pairs separated by a colon:

- **Target Namespace** — The target name space identifier, typically a Universal Resource Name (URN)
- **Status** — Indicates if the Web service is enabled or disabled for client access
- **AppServerInfo** — The AppServer connection information, expressed as a URL
- **Session Model** — Indicates if the Web service requires an AppServer configured with session-managed or session-free AppServer operating modes
- **Style/Use** — SOAP message style (RPC or Document)

In addition, the status includes the a list of current application properties.

## Viewing Web service statistics

You can view (and reset) run-time statistics for a selected Web service. Each Web service creates and begins accumulating these statistics (Start Time) from the time that its parent WSA instance starts up (Creation Time). You can reset the Web service statistics at any point while the WSA instance is running. A reset clears and begins accumulating the statistics again with a new Start Time.



### To display the run-time statistics for a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service whose statistics you want to view is deployed. The **Details** page for that instance appears.
3. Choose the Web service name from the dropdown list, and then click **Select**.

4. Under **Deployed Web Services**, click **Statistics**. A read-only statistics presentation with the following information appears:
  - **Number of Requests** — Total requests handled by the Web service
  - **Number of Active Requests** — Requests currently being serviced by the Web service
  - **Number of SOAP Faults** — Errors returned by the Web service application
  - **Number of Objects Not Found** — Calls to unidentified Web service methods
  - **Number of Errors** — Total errors returned by the Web service, with error counts broken out at the bottom of the list for each of several error categories when the total errors in each category are greater than 0
  - **Number of AppServer Connections** — Total AppServer connections in the connection pool for this Web service
  - **Number of AppObjects** — Total AppObjects registered for all clients of this Web service (only one AppObject is shared by all clients of a session-free Web service; no AppObjects are ever shared for a session-managed Web service)
  - **Number of SubAppObjects** — Total SubAppObjects registered for all clients of this Web service (SubAppObjects never shared)
  - **Number of ProcObjects** — Total ProcObjects registered for all clients of this Web service (ProcObjects never shared)
5. To reset the statistics and start accumulating from zero, click **Reset** and confirm that you want to restart the statistics.
6. Click **OK** on the confirmation message.

## Updating a deployed Web service

You can update the definition and deployment information for a deployed Web service at any time.

---

**Caution:** You can freely update the definition and deployment information for a deployed Web service during development. However, once you deploy and enable a Web service for client access under production conditions, avoid making any changes to this information, as client implementations depend on its stability.

---

After production deployment, to make the same Web service available using different information (for example, to add a new operation or use a different SOAP Message Style), deploy a new Web service with the new information, using a different Web service name in the same WSA context. You can also deploy the Web service to a different WSA context.

## Updating the definition and deployment information

You must disable a Web service before you can update it. You can verify if the Web service is disabled by checking its status.

Once you enable the Web Service for client access, avoid making any changes to this deployment information, as client implementations depend on its stability. To make the same Web service available using different information (for example, to add a new operation or use a different **WSDL Style/Use**), deploy a new Web service for the application with a different name or in a different WSA context, and with the new information you want to use.



### To update a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service you want to update is currently deployed. The **Details** page for that instance appears.
3. Click **Update**.
4. Type the path and name of the WSM file for the Web service that you want to update.
5. Click **Submit** to update deployment information. The Web service name for an update is read-only and preset from the initial deployment. Initial values for other fields are set from existing information in the new WSM file, and you can change them as necessary. The fields that you can update include:
  - **Web Service Namespace** — Any value you choose to uniquely qualify the names for operations and parameters used to define the Web service.
  - **SOAP Action** — A string that the client application is required to place in the SOAPAction HTTP header when accessing a Web service hosted by the WSA instance. The SOAPAction HTTP header is required for all HTTP messages that carry SOAP messages and is used by intervening security servers (such as firewalls) to determine if each HTTP message is allowed to pass through to its destination. The default is a blank string, but can be any string required by the intervening security servers on the network.
  - **Append to SOAP Action** — Indicates whether or not to append the specified SOAP Action value to any other SOAP Action values generated for the Web service SOAP messages by Web service clients.
  - **WSDL Style/Use** — Specifies the SOAP format to use when sending or receiving SOAP messages for this Web service. The value that you choose is dependent on what your anticipated Web service clients can support.
6. Click **Submit**. A confirmation message appears.

You might need to document additional information about the updated Web service for use by Web service client developers. For more information, see the chapter on client requirements in *OpenEdge Development: Web Services* in the Product Documentation section of the Progress Software Developers Network (PSDN) Web site.

## Setting Web service application properties

You can view and set the value of each property for a deployed Web service, or you can reset the values of all properties to the current Web Service defaults (stored in the `default.props` file for the WSA instance where the Web service is deployed).

You must disable a Web service before you can set most properties for it. You can verify if the Web service is disabled by checking its status. The only properties that you can set for an enabled Web service are:

- `serviceFaultLevel`
- `serviceLoggingLevel`

If you set other properties while the Web service is enabled, the property value changes take effect only after you disable and then enable the Web service again.

---

**Note:** Unlike WSA instance properties, the values for Web service properties are not stored in the `ubroker.properties` file. They are stored in the `WebServiceFriendlyName.props` XML file located in the Web application servlet directory for the WSA instance where the Web service is deployed.

---



### To view and set Web service properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service whose properties you want to set is deployed. The **Details** page for that instance appears.
3. Choose the Web service name from the dropdown list, and click **Select**. The list of application properties appears. For details on the properties that can appear in this list, see [Appendix A, “Reference to OpenEdge Web Service Properties.”](#)

---

**Note:** The actual list of properties displayed is a subset of all the properties, and changes depending on whether the session model of the Web service is session managed or session free. To verify the session model of a Web service, check its status. For more information, see the “[Viewing the status of a Web service](#)” section on page 6–26.

---

4. Click **Edit**.
5. Type the new value in the available field. For logical values, entering any value other than true results in a setting of false.
6. Click **Submit** when you finish. The changes are then written to the `WebServiceFriendlyName.props` file for the selected Web service.

## Undeploying a Web service

You can undeploy a deployed Web service, which removes its deployment information, its properties, and other settings from the WSA instance context.

You must disable a Web service before you can undeploy it. You can verify if the Web service is disabled by checking its status.



### To undeploy a Web service:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. Expand the **Web Services Adapter** folder, and select the instance where the Web service you want to undeploy is currently deployed. The **Details** page for that instance appears.
3. Choose the Web service name from the dropdown list, and click **Select**.
4. Click **Undeploy**.

If the Web service is already disabled, it is now undeployed from the WSA instance context and removed from the management console. If the Web service is still enabled, an error appears indicating that you must disable the Web service before undeploying it.

The changes are then written to the `WebServiceFriendlyName.props` file for the selected Web service.



---

## Configuring WebSpeed Messengers and WebSpeed Transaction Servers

---

You can configure property settings for WebSpeed Messengers and examine their log files in OpenEdge Management and OpenEdge Explorer.

You can configure WebSpeed Transaction Server property settings and add new WebSpeed instances in OpenEdge Management and OpenEdge Explorer. You can also start, stop, and delete WebSpeed instances, as well as view their status and log files.

This chapter provides information about WebSpeed Messenger and WebSpeed Transaction Server configuration, as described in the following topics:

- [WebSpeed Messenger administration](#)
- [WebSpeed Transaction Server administration](#)

## WebSpeed Messenger administration

The WebSpeed Messenger resides on your Web server machine. It picks up incoming application service requests from WebSpeed clients and directs them to a WebSpeed broker that supports that application service. The Messenger is either a CGI program, or an ISAPI or NSAPI process.

There are four different WebSpeed Messengers:

- **CGIIP Messenger** — This Messenger runs on almost all Web servers, but tends to have the slowest response times.
- **WSASP Messenger** — This Messenger is used to call WebSpeed applications from a Microsoft Active Server Page. It cannot coexist with any other Messenger on your Web server.
- **WSISA Messenger** — This Messenger runs on Microsoft IIS Web servers.
- **WSNSA Messenger** — This Messenger runs on Netscape Web servers.

You cannot create or delete WebSpeed Messengers from the management console. You can, however, edit the Messenger's properties, enable or disable the Messenger, and examine the Messenger's log file from the console.

## Configuring a WebSpeed Messenger

You can view or edit properties in the following categories for a WebSpeed Messenger in the management console:

- General
- Controlling NameServer
- Logging Setting
- SSL
- Advanced

You cannot create or delete WebSpeed Messengers from the management console. You can, however, edit the Messenger's properties, enable or disable the Messenger, and examine the Messenger's log file from the console.

Procedures for viewing or editing the configuration are described in the sections that follow.

### Viewing or modifying WebSpeed Messenger properties

You can view or modify General, Controlling NameServer, Logging Setting, SSL and Advanced properties for each WebSpeed Messenger.



**To view or modify WebSpeed Messenger properties:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.

3. Expand the **Messengers** category, and select a messenger. The **Details** page for that messenger appears in the detail frame.
4. Click **Configuration**.

From this page, you can do the following:

- View a read-only display of the Messenger properties as described in the [Table 7-1](#) through [Table 7-6](#). All of these tables follow this procedure.
- Click **Edit** to modify the broker properties. For details, see the “[Editing a WebSpeed Messenger configuration](#)” section on page 7-7.

**Table 7-1: WebSpeed Messenger General properties**

Property	Description
<b>Working directory</b>	The Messenger's working directory, where it stores temporary files and the Messenger log file
<b>Messenger executable filename</b>	The full path where the Messenger executable file resides

**Table 7-2: WebSpeed Messenger Controlling NameServer properties (1 of 2)**

Property	Description
<b>Register NameServer</b>	
<b>Register with NameServer</b>	Register with the controlling NameServer.
<b>NameServer Information</b>	
<b>Controlling NameServer</b>	The NameServer with which the WebSpeed Messenger registers. The WebSpeed Messenger then uses the NameServer to find a WebSpeed broker that supports the current application service. (If you did not choose to register the messenger with a NameServer, the field is dimmed and unavailable.)

**Table 7–2: WebSpeed Messenger Controlling NameServer properties** (2 of 2)

Property	Description
<b>Minimum NameServer client port</b>  and  <b>Maximum NameServer client port</b>	<p>These properties are used to specify a range of UDP ports for the messenger instance. From this range, a messenger instance can select which UDP port to use when communicating with the NameServer. To limit the range, network administrators can specify a Minimum and a Maximum value.</p> <p>Specifying a range using Minimum and Maximum allows network administrators to limit the number of UDP ports needed to be open in the firewall when the NameServer client (in this case, the messenger) and the NameServer are separated by a firewall. The value for Minimum and Maximum can be zero (0) or a number between 1024 and 65535 inclusive.</p> <p>The Minimum value must be less than or equal to the Maximum value. If both the Minimum and Maximum values are set to zero (0), then a random port number within the implicit fixed range is dynamically assigned. How the port is assigned depends on the platform on which the messenger instance is running. If both Minimum and Maximum values are set to the same valid port number, then this UDP port number is used when communicating with the NameServer.</p> <p>The default value for both Minimum and Maximum is 0.</p>
<b>WebSpeed Transaction Server Location</b>	
<b>WebSpeed transaction server host</b>	The WebSpeed Transaction Server host location.
<b>WebSpeed transaction server port</b>	The WebSpeed Transaction Server port location.

## Logging setting properties

For complete information about these logging settings, see *OpenEdge Development: Debugging and Troubleshooting* in the Product Documentation section of the Progress Software Developer's Network (PSDN) Web site.

**Table 7–3: WebSpeed Messenger Logging Setting properties** (1 of 2)

Property	Description
<b>Logging level</b>	<p>A value that specifies the amount of information to be written to the Messenger log. Select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Logging entry types</b>	<p>The one valid entry type for WebSpeed Messenger logging is <code>MsgrTrace</code>, which turns on logging for WebSpeed Messengers. The information logged depends on which Messenger is running and the logging level specified. For example:</p> <ul style="list-style-type: none"> <li>• <b>Basic</b> — For CGIIP, logs the start and end of the request, and the values of the <code>CONTENT_LENGTH</code>, <code>PATH_INFO</code> and <code>QUERY_STRING</code> variables. For other Messengers, logs the environment variables.</li> <li>• <b>Verbose</b> — Same as <b>Basic</b> but also includes the CGIIP environment variables and the environment variables in binary format for other Messengers.</li> <li>• <b>Extended</b> — Same as <b>Verbose</b>, but also includes the environment variables in binary format for CGIIP.</li> </ul> <p>You can specify a logging level on the entry type. For example: <b>MsgrTrace:3</b></p>
<b>Log file threshold size</b>	<p>A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.</p>

**Table 7–3: WebSpeed Messenger Logging Setting properties** (2 of 2)

Property	Description
<b>Maximum number of log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of server log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p style="text-align: center;">filename.#####.extension</p>

**Table 7–4: WebSpeed Messenger SSL properties**

Property	Description
<b>Enable SSL client connections</b>	If checked, this property directs that the connection to the WebSpeed Transaction Server use SSL tunneling.
<b>Disable client verification of SSL host name</b>	If checked, this property turns off host verification for an SSL connection. If cleared, the client compares the host name for the WebSpeed Transaction Server specified by Host Name (or as returned by any Controlling NameServer) with the Common Name specified in the server digital certificate, and raises an error if they do not match. With this parameter specified, the client never raises the error.
<b>Disable SSL session reuse</b>	If this property is checked, the connection does not reuse the SSL session ID when reconnecting to the same WebSpeed Transaction Server.

**Table 7–5: WebSpeed Messenger Advanced properties** (1 of 2)

Property	Description
<b>Internal administration command</b>	When this property is selected, the Messenger allows access to the WSMAdmin Web page. The WSMAdmin Web page provides links to customize error messages, enable or disable session logging, and validate the traffic flow between the Messenger and the WebSpeed broker/agents.

**Table 7–5: WebSpeed Messenger Advanced properties** (2 of 2)

Property	Description
<b>Session connection ID</b>	When this property is selected, WebSpeed Session Connection IDs are enabled.
<b>Authorized IP list</b>	The list of all IP addresses that are allowed to access the WSMAdmin Web page, if it has been enabled. If you add any addresses to this list, then only requests coming from addresses that are on the list will reach the WSMAdmin Web page. This gives you the ability to tailor access to the WSMAdmin Web page, so that only users connecting from certain sites can access the administration functions available on that page.

## Editing a WebSpeed Messenger configuration

You can edit the configuration of a WebSpeed Messenger.



### To edit a WebSpeed Messenger configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the Messenger whose configuration you want to edit. The **Details** page for that messenger appears in the detail frame.
3. Click **Configuration**. The messenger's **Configuration** page appears.
4. Click **Edit**, and make the changes to the properties. (For details about the properties, see [Table 7–1](#), [Table 7–2](#), [Table 7–3](#), [Table 7–4](#), and [Table 7–5](#).)
5. Click **Save**.

## Viewing the WebSpeed Messenger log file

You can view the log file for a WebSpeed Messenger by using the log file viewer. The log file viewer allows you to examine the log file through an HTML interface.



### To access the log file viewer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the messenger. The **Details** page for that messenger appears in the detail frame.
3. Click **Log File Viewer**.

You can use with the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display 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 at 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, regardless of 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 **Previous** 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**.
- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.
- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.



# WebSpeed Transaction Server administration

With the WebSpeed Transaction Server, you can make WebSpeed applications available over the Web, an Intranet, or an Extranet.

You can use OpenEdge Management or OpenEdge Explorer to configure new WebSpeed Transaction Server instances and administer the configurations of existing instances. The WebSpeed Transaction Server installation provides one sample WebSpeed Transaction Server (wsbroker1). You can use this sample component as a template for creating and configuring additional WebSpeed Transaction Servers. Each WebSpeed Transaction Server is called an instance.

## Working with the WebSpeed Transaction Server

Each WebSpeed Transaction Server consists of a WebSpeed broker and one or more WebSpeed agents. A WebSpeed agent is a single ABL Virtual Machine (AVM) instance running on a WebSpeed Transaction Server. Each WebSpeed agent executes ABL procedures on behalf of WebSpeed clients.

To work with a WebSpeed Transaction Server instance, double-click the **WebSpeed** folder from the management console's list frame. There is one pre-defined default WebSpeed Transaction Server (wsbroker1).

You can perform the following WebSpeed Transaction Server actions:

- [Configuring a WebSpeed Transaction Server](#)
- [Editing a WebSpeed Transaction Server configuration](#)
- [Creating a WebSpeed broker](#)
- [Starting and stopping a WebSpeed broker](#)
- [Adding or trimming WebSpeed agents](#)
- [Deleting a WebSpeed broker](#)

## Configuring a WebSpeed Transaction Server

You can view or edit the following configuration properties of a WebSpeed Transaction Server broker from the management console:

- Broker
- Agent
- SSL
- Messaging
- Environment variables

You can also create a new WebSpeed Transaction Server broker, delete a broker you no longer need, or view the broker's log files.

Each of these tasks is described in the sections that follow.

## Viewing or modifying WebSpeed Transaction Server broker properties

You can view or modify the following WebSpeed Transaction Server Broker properties: general properties, owner information, controlling NameServer information, AppService Name List information, logging setting details, and advanced features.



### To view or modify WebSpeed Transaction Server broker properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **WebSpeed** category, and select a WebSpeed Transaction Server Broker. The **Details** page for the broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **WebSpeed Configuration** page opens with that Broker's properties in focus.

From this page, you can do the following:

- View a read-only display of the WebSpeed broker configuration properties as described in [Table 7–6](#) through [Table 7–11](#). All these tables follow this procedure.
- Click **Edit** to modify the broker properties. For details, see the “[Editing a WebSpeed Transaction Server configuration](#)” section on page 7–27.

**Table 7–6: WebSpeed Broker General properties**

Property	Description
<b>Auto start</b>	Indicates whether or not the server automatically starts when the controlling AdminServer starts.
<b>Port number</b>	The number of the TCP/IP port that the server broker listens on.
<b>Working directory</b>	Your working directory. The default is the working directory set during OpenEdge installation.
<b>Operating mode</b>	The operating mode for this server.

**Table 7–7: WebSpeed Broker Owner Information properties** (1 of 2)

Property	Description
<b>Username</b>	The username of an account that has system-administrative rights.
<b>Group name</b>	(UNIX only) The group name on UNIX.

**Table 7–7: WebSpeed Broker Owner Information properties** (2 of 2)

Property	Description
<b>Password</b>	The username account password.
<b>Password confirm</b>	Confirmation of the username account password. This field appears only if you are editing the configuration.  Note that this field does not appear if the AdminServer is running on a non-Windows host machine.

**Table 7–8: WebSpeed Broker Controlling NameServer properties**

Property	Description
<b>Register with NameServer</b>	Indicates whether or not to register the broker with a controlling NameServer.
<b>Controlling NameServer</b>	The name of the NameServer with which this broker registers. (If you did not select to register the broker with a NameServer, the field is unavailable.)
<b>Registration mode</b>	How the broker specifies its hostname if it is registering with a controlling NameServer.  This hostname information is passed onto a client application when it attempts to connect to an application service that the broker supports. The choices are: <ul style="list-style-type: none"> <li>• <b>Register-IP</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li> <li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li> <li>• <b>Register-HostName</b> — The broker registers with the values specified in the HostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li> </ul>
<b>Registration host name</b>	The name of the non-local host.

**Table 7–9: WebSpeed Broker AppService Name List properties**

Property	Description
<b>Application service names</b>	A list to which you can add or delete application service names.
<b>Supports default service</b>	Indicates whether the AppServer broker supports the default service.

**Table 7-10: WebSpeed Broker Logging Setting properties***(1 of 3)*

Property	Description
<b>Broker log filename</b>	The broker log filename.
<b>Broker logging level</b>	<p>A value that specifies the amount of information that is written to the broker log. The following values are available:</p> <ul style="list-style-type: none"><li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li><li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li><li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li><li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li><li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li></ul>
<b>Append to broker log file</b>	Indicates whether or not a new broker log file should be created when the broker is started.

Table 7–10: WebSpeed Broker Logging Setting properties

(2 of 3)

Property	Description
<b>Broker logging entry types</b>	<p>A comma-separated list of the following valid entry types for agent logging:</p> <ul style="list-style-type: none"> <li>• <b>UBroker.AutoTrim</b> — Logs messages about automatic agent thread trimming by the broker based on the Auto trim timeout property. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.Basic</b> — Logs messages about broker and agent startup and shutdown at the following logging levels: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs NameServer registration and connections from clients.</li> <li>– <b>Verbose</b> — Logs information about all the property values read from the <code>ubroker.properties</code> file for that broker instance, and more information on the connection from clients.</li> </ul> </li> <li>• <b>UBroker.ThreadPool</b> — Logs messages about the pool of threads managed by the broker. These threads are used to control the client requests and the agent processes. This entry type logs messages when adding, removing, and communicating with these threads. Information is logged at <b>Basic</b> and higher.</li> </ul> <p>Turn on the following logging entry types only at the request of Progress Software Corporation Technical Support:</p> <ul style="list-style-type: none"> <li>• <b>UBroker.ClientFSM</b> — Logs messages about the state of the client thread as it processes the requests, from the moment it receives a connection request to the end of the request. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerFSM</b> — Logs messages about the state of the agent thread as it processes the requests, from the moment it receives a connection request to the end of the request. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMsgStream</b> — Logs messages about messages exchanged between the broker and client threads in binary format. Information is logged at <b>Basic</b> and higher.</li> </ul>

**Table 7–10: WebSpeed Broker Logging Setting properties**

(3 of 3)

Property	Description
<b>Broker logging entry types</b> <i>(continued)</i>	<ul style="list-style-type: none"> <li>• <b>UBroker.ServerMsgStream</b> — Logs messages about messages exchanged between the broker and agent threads in binary format. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMsgQueue</b> — Logs messages about the client thread as it processes queued messages. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerMsgQueue</b> — Logs messages about the agent thread as it processes queued message. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ClientMemTrace</b> — Traces messages being processed by the client threads. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.ServerMemTrace</b> — Traces messages being processed by the agent threads. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.Stats</b> — Logs statistics about the number of requests, plus maximum, minimum, and average request wait and duration times. Information is logged at <b>Basic</b> and higher.</li> <li>• <b>UBroker.All</b> — Logs all UBroker entry types.</li> </ul> <p>You can specify a separate logging level for each log entry type and use a wildcard (*) to specify multiple entry types. For example:</p> <pre>UBroker.Basic:3,UBroker.ThreadPool:4,UBroker.Client*</pre>
<b>Broker log file threshold size</b>	A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the log file size other than what the operating system imposes.
<b>Maximum number of broker log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of broker log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Broker log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <pre>filename.#####.extension</pre>

**Table 7–11: WebSpeed Broker Advanced Features properties** (1 of 2)

Property	Description
<b>Maximum client instances</b>	<p>The maximum number of client connections that the broker can support concurrently.</p> <p>The default is 512, which is high enough to ensure that the number of client connections is virtually unlimited. Realistically, however, the system-level resources required to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you might need to reduce the maximum number of client connections to a more reasonable value.</p> <p>Note, however, that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads will alleviate the problem.</p>
<b>Priority weight (0-100)</b>	<p>An integer value between 0 and 100 that influences the share of the workload that the selected AppServer receives. The larger the value, the heavier the load that is distributed to the server. The NameServer distributes client requests across all WebSpeed Transaction Servers that have the same application service in proportion to the <b>Priority weight</b> value.</p> <p>The default is 0.</p>
<b>Registration retry (in seconds)</b>	<p>After registering with its controlling NameServer, the WebSpeed Transaction Server periodically sends "keep-alive" messages to the NameServer to let the NameServer know that it is still active. The <b>Registration retry</b> value is the number of seconds that pass between "keep-alive" messages.</p> <p>The default is 30.</p>
<b>Server startup timeout</b>	<p>Specifies the amount of time, in seconds, that the broker waits for an active agent to become available before starting a new agent process.</p> <p>The default is 3.</p>
<b>Request timeout</b>	<p>Specifies the amount of time, in seconds, that the broker waits for an agent to become available for processing a request.</p> <p>The broker waits for this period only if the <b>Maximum servers</b> setting has been reached. After the timeout has expired, the client receives a "no servers available" error message.</p> <p>The default is 15.</p>

**Table 7–11: WebSpeed Broker Advanced Features properties**

(2 of 2)

Property	Description
<b>Auto trim timeout</b>	<p>The amount of time, in seconds, that the broker waits before automatically trimming the number of running agents.</p> <p>The broker keeps track of the maximum number of agents that are simultaneously busy during the interval you specify. At the end of the interval, the broker attempts to trim the number of agents to match either the <b>Maximum servers</b> for the interval, or the <b>Minimum servers</b>, whichever is greater.</p> <p>You can disable this feature by setting the timeout to zero (0). The default <b>Server Auto-trim Interval</b> is 1800 seconds.</p> <p>Progress Software recommends that you use this default setting to avoid unnecessary process management that might result from frequent starting and trimming of agents.</p>
<b>TCP/IP version</b>	<p>Internet Protocol for network communication:</p> <ul style="list-style-type: none"> <li>• <b>IPv4</b> — Accepts only IPv4 connections.</li> <li>• <b>IPv6</b> — Allows IPv4 and IPv6 connections using mapped address (where supported).</li> </ul> <p>The default is IPv4.</p> <p>You must also set the <b>jvmArgs</b> property in the <code>ubroker.properties</code> file to override the default behavior of the JVM for IPv6 connections.</p> <pre>jvmArgs=-Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true</pre>
<b>jvmargs</b>	Java System Properties.

### Viewing or modifying WebSpeed Agent properties

You can set the following agent properties for a WebSpeed Broker: general properties, logging setting details, pool range, and advanced features.



#### To view or modify WebSpeed Agent properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **WebSpeed** category, and select a WebSpeed broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **WebSpeed Configuration** page opens with the **Broker** properties in focus.



5. Click the **Agent** tab.

From this page, you can do the following:

- View a read-only display of the WebSpeed Agent properties as described in [Table 7–12](#) through [Table 7–15](#). All of these tables follow this procedure.
- Click **Edit** to modify the broker properties. For details, see the “[Editing a WebSpeed Transaction Server configuration](#)” section on page 7–27.

**Table 7–12: WebSpeed Agent General properties**

Property	Description
<b>Server executable file</b>	Either the default agent executable pathname or a different agent executable file of your choosing. You generally only need to specify a nondefault value if you have generated a new executable using the OEBUILD utility.
<b>Server startup parameters</b>	The OpenEdge startup parameters that you want to specify to start each agent.
<b>PROPATH</b>	The search path that agents use to locate ABL procedures that they execute. Specify a list of directory pathnames in the form of an OpenEdge PROPATH. Make sure that you have copied all of your WebSpeed Transaction Server procedures to one or more of the directories in the PROPATH.
<b>Minimum port number</b>	The minimum TCP/IP port number from a range that each agent process can listen on. When each agent starts, it allocates a port that is not being used from the specified range, up to the <b>Maximum port number</b> setting.
<b>Maximum port number</b>	The maximum TCP/IP port number from a range that each agent can listen on. When each agent starts, it allocates a port that is not being used from the specified range, down to the <b>Minimum port number</b> setting.
<b>Server application mode</b>	Development or Production

**Table 7–13: WebSpeed Agent Logging Setting properties***(1 of 4)*

Property	Description
Server log filename	The server log file name.
Server logging level	<p>A value that specifies the amount of information to be written to the agent log. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"><li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li><li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li><li>• <b>Basic</b> —Server logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li><li>• <b>Verbose</b> —Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li><li>• <b>Extended</b> —Server logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li></ul>
Append to server log file	Indicates whether a new agent log file should be created when the WebSpeed Transaction Server is started.

Table 7–13: WebSpeed Agent Logging Setting properties

(2 of 4)

Property	Description
Server logging entry types	<p>A comma-separated list of valid entry types for agent logging:</p> <ul style="list-style-type: none"> <li>• <b>4GLTrace</b> — Turns on the logging for the execution of the following ABL statements: RUN, FUNCTION (user-defined functions), PUBLISH and SUBSCRIBE. Logs information at <b>Basic</b> or higher.</li> <li>• <b>ASDefault</b> — Combines the <b>ASPlumbing</b> and <b>DB.Connects</b> log entry types. This is the default value for AppServer agents.</li> <li>• <b>ASPlumbing</b> — Turns on logging for different actions depending on the logging level specified: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs messages when an AppServer agent starts and stops, and when client processes connect and disconnect from the AppServer agent.</li> <li>– <b>Verbose</b> — Same as <b>Basic</b>, plus logging of the execution of the top level procedure (that is, the procedure that the client process asks the AppServer agent to execute on its behalf, but not any procedure that the top level procedure executes).</li> <li>– <b>Extended</b> — Same as <b>Verbose</b>, plus messages about agent state, and status messages about read and write socket operations. Do not turn this on unless instructed to do so by Progress Software Technical Support.</li> </ul> </li> <li>• <b>DB.Connects</b> — Turns on logging of database connections and disconnections. The messages include database name and user ID number. Logs information at <b>Basic</b> or higher.</li> <li>• <b>DynObjects.DB</b> — Turns on logging for dynamic database-related objects (for example, TEMP-TABLE or DATASET) as they are created and destroyed. Logs information at <b>Basic</b> or <b>Verbose</b>.</li> <li>• <b>DynObjects.XML</b> — Turns on logging for dynamic XML-related objects (for example, x-document or x-noderef) as they are created and destroyed. Logs information at <b>Basic</b> or <b>Verbose</b>.</li> </ul>

**Table 7–13: WebSpeed Agent Logging Setting properties**

(3 of 4)

Property	Description
<b>Server logging entry types</b> <i>(continued)</i>	<ul style="list-style-type: none"> <li>• <b>DynObjects.Other</b> — Turns on logging for dynamic objects that do not match the other dynamic object categories (for example, procedure or server socket) as they are created and destroyed. Logs information at <b>Basic</b> or <b>Verbose</b>.</li> <li>• <b>ProEvents.UI.Char</b> — Turns on logging of keystroke events for printable characters. Logs information at <b>Basic</b> or higher.</li> <li>• <b>ProEvents.UI.Command</b> — Turns on logging of keystroke events for nonprintable characters at the following logging levels: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Logs nonprintable keystrokes and a subset of Windows GUI events.</li> <li>– <b>Verbose</b> — Logs nonprintable keystrokes and all Windows GUI events.</li> </ul> </li> <li>• <b>ProEvents.Other</b> — Turns on logging of COM, asynchronous, and server socket events. Logs information at <b>Basic</b> or higher, beginning with events for which you have written triggers.</li> <li>• <b>QryInfo</b> — Turns on database query information logging at the following logging levels, with the same amount of information logged for each level: <ul style="list-style-type: none"> <li>– <b>Basic</b> — Log static queries and initial dynamic queries.</li> <li>– <b>Verbose</b> — Logs static queries and multiple instances of dynamic queries.</li> </ul> </li> <li>• <b>SAX</b> — Turns on logging for various stages of execution using the ABL SAX parser.</li> </ul> <p>You can specify a separate logging level for each log entry type, and you can use a wildcard (*) to specify multiple entry types. For example:</p> <pre>4GLTrace:3,ASPLubing:2,DynObjects.*</pre>
<b>Server log file threshold size</b>	A value of 0 or a value between 500,000 and 2,147,483,647, where 0 means there is no limit on the agent log file size other than what the operating system imposes.

**Table 7–13: WebSpeed Agent Logging Setting properties**

(4 of 4)

Property	Description
<b>Maximum number of server log files</b>	<p>The number of rolled-over log files to keep. The value can be 0 or a value between 2 and 999999, where 0 means there is no limit on the number of agent log files to keep. The specified number represents the maximum total number of log files to keep on disk at any time, including the current log file being written to.</p> <p>When the file becomes equal to or greater than the <b>Server log file threshold size</b>, the client process renames it and creates a new log file. The file is renamed as follows, where ##### is a number starting at 000001 and increasing to 999999, after which it rolls back over to 000001:</p> <p><i>filename.#####.extension</i></p>
<b>Server watch dog interval</b>	<p>The interval (in seconds) for the server log file watchdog thread. If the <b>Server log file threshold size</b> property is specified, the broker will roll over the server log file when its size is larger than the logThreshold value.</p>

**Table 7–14: WebSpeed Agent Pool Range properties**

Property	Description
<b>Initial number of servers to start</b>	The number of agents you want the broker to start initially.
<b>Minimum servers</b>	The minimum number of agents before the broker starts additional agents. If you trim the number of agents below this value, the server starts any additional agents needed to maintain the specified minimum when the next client connects.
<b>Maximum servers</b>	The maximum number of agents that this server can have running at the same time.

**Table 7-15: WebSpeed Agent Advanced Features properties***(1 of 2)*

Property	Description
<b>Application URL</b>	A server-relative URL (starting with a "\") to the path of the application. If the Web server redirects an incoming URL such that the SCRIPT_NAME and PATH_INFO variables do not refer to the application, setting this property allows an application to create self-referencing URLs when SCRIPT_NAME and/or PATH_INFO are not correct. If this option is set, the value of the applicationURL global variable available to applications is set to this value instead of a combination of SCRIPT_NAME and PATH_INFO.
<b>Default cookie path</b>	A server-relative URL (starting with a "\") to use for all cookies if not explicitly specified in the set-cookie() or delete-cookie() functions. The default is the value of the AppURL global variable.
<b>Default cookie domain</b>	Domain to send cookies. The default is blank, causing the Web browser to set cookies only for the current host. If cookies must be available to multiple hosts in the same domain by default, set that domain name here; for instance, .progress.com. Unless a value is specified for the domain option for set-cookie() and delete-cookie() functions, this value is used.
<b>File upload directory</b>	Specifies a directory where WebSpeed servers can upload files. Refer to the ubroker.properties file for an HTML snippet that shows how to use this feature.  By default this capability is disabled. To enable it, specify an upload directory. Only text file upload is supported, not binary.
<b>Binary upload max size</b>	Specifies the maximum size (in bytes) of binary files that WebSpeed handles. The application can access the contents of the binary files through the ABL language. Note that this property is not related to the <b>File upload directory</b> property; therefore, WebSpeed will not upload the file to a directory. You can set any of the following options: <ul style="list-style-type: none"> <li>• <b>0</b> — Binary upload is disabled.</li> <li>• <b>-1</b> — No maximum size of a binary upload; binary upload enabled.</li> <li>• Any non-zero value greater than 0 — Maximum size of binary upload.</li> </ul>
<b>Debug mode</b>	Whether the debug mode is enabled, disabled, or using the default.

**Table 7–15: WebSpeed Agent Advanced Features properties** (2 of 2)

Property	Description
<b>4GL debugger enabled</b>	Whether the debugger is enabled or disabled: <ul style="list-style-type: none"> <li><b>Disabled</b> — Specifies ?debug=on in the URL has no effect.</li> <li><b>Enabled</b> — Allows debug information if ?debug=on is in the URL.</li> <li>&lt;Blank&gt; — Defers behavior to <code>svrAppMode</code> (<b>Disabled</b> in Production, <b>Enabled</b> in Development)</li> </ul>
<b>Startup parameters</b>	The parameters to pass the <b>Startup</b> procedure when it executes. The <b>Startup</b> procedure takes a character string as an input parameter. For example: <pre>DEFINE INPUT PARAMETER startup-data AS CHARACTER NO-UNDO.</pre> <p>You can set this parameter to any arbitrary value. If you do not specify a parameter in this field, the parameter is set to the unknown value (?) when the WebSpeed Transaction Server agent executes the <b>Startup</b> procedure.</p>

### Viewing or modifying WebSpeed SSL properties

You can set properties in the following WebSpeed Broker SSL categories: general properties and advanced features.



#### To view or modify WebSpeed SSL properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **WebSpeed** category, and select a WebSpeed broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **WebSpeed Configuration** page opens with the **Broker** properties in focus.
5. Click the **SSL** tab.

From this page, you can do the following:

- View a read-only display of the WebSpeed SSL properties as described in the [Table 7–16](#) and [Table 7–17](#). Both of these tables follow this procedure.
- Click **Edit** to modify the SSL properties. For details, see the “[Editing a WebSpeed Transaction Server configuration](#)” section on page 7–27.

**Table 7–16: WebSpeed SSL General properties**

Property	Description
<b>Enable SSL client connections</b>	If checked, this property specifies that all connections to this WebSpeed Transaction Server must use SSL tunneling.
<b>Private key/Digital certificate alias name</b>	The alias name within the OpenEdge keystore of the private key and digital certificate entry to use to authenticate all connections to this WebSpeed Transaction Server. If not otherwise selected, the WebSpeed Transaction Server uses the default_server server certificate alias.
<b>Password to access the key/certificate</b>	<p>The password to use for accessing the private key and digital certificate. You must specify a password when you specify the select an OpenEdge keystore alias name.</p> <p><b>Note:</b> Your password is encrypted in the <code>ubroker.properties</code> file. If you set the password in <code>ubroker.properties</code> manually, you must specify the password as an encrypted value. You can obtain this value using the <code>genpassword</code> utility located in the <code>bin</code> directory of your OpenEdge installation.</p> <p>If you use the default_server server certificate, it also has a default password that you do not need to specify.</p>
<b>Password confirm</b>	<p>Confirmation of the username account password. This field appears only if you are editing the configuration.</p> <p>Note that this field does not appear if the AdminServer is running on a non-Windows host machine.</p>

**Table 7–17: WebSpeed SSL Advanced Features properties**

Property	Description
<b>Disable SSL session caching</b>	If this property is checked, caching for the SSL client session is disabled.
<b>SSL session cache timeout</b>	<p>The length of time, in seconds, that an SSL client session is held in the session cache, during which an SSL client can resume its session.</p> <p>The default is 180.</p>



## Viewing or modifying WebSpeed Messaging properties

You can view or modify several WebSpeed Messaging properties.



### To view or modify WebSpeed Messaging properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **WebSpeed** category, and select a WebSpeed broker. The **Details** page for that broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **WebSpeed Configuration** page opens with the **Broker** properties in focus.
5. Click the **Messaging** tab.

From this page, you can do the following:

- View a read-only display of the WebSpeed Messaging properties as described in [Table 7–18](#), which follows this procedure.
- Click **Edit** to modify the WebSpeed Messaging properties. For details, see the [“Editing a WebSpeed Transaction Server configuration”](#) section on page 7–27.

**Table 7–18: WebSpeed Messaging properties**

(1 of 2)

Property	Description
<b>SonicMQ ServerConnect enabled</b>	Allows the asbroker or wsbroker agent to start a personal SonicMQ Adapter instead of going to a separate one running elsewhere.
<b>SonicMQ ServerConnect broker log filename</b>	A valid path and file name.

Table 7–18: WebSpeed Messaging properties

(2 of 2)

Property	Description
<b>Broker logging level</b>	<p>A value that specifies the amount of information to be written to the SonicMQ ServerConnect broker log. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Broker logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Broker logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	Indicates whether a new broker log file should be created, even if the broker log file already exists, when the SonicMQ ServerConnect broker is started.
<b>SonicMQ ServerConnect server log filename</b>	A valid path and file name.
<b>Server logging level</b>	<p>A value that specifies the amount of information to be written to the SonicMQ ServerConnect server log. Each logging level name has the indicated numeric value. You can select from the following values in the drop-down list:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Broker logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Broker logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Broker logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Append to server log file</b>	Indicates whether a new server log file should be created, even if the broker log file already exists, when the SonicMQ ServerConnect instance is started.

## Setting or deleting WebSpeed environment variables

You can set or delete environment variables. For WebSpeed, environment variables that appear in the list are scoped to:

- The process in which the WebSpeed Transaction Server broker executes
- Every WebSpeed Transaction Server agent that the WebSpeed Transaction Server broker starts



### To set or delete WebSpeed environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker whose environment variables you want to set. The **Details** page for that broker appears in the detail frame.
3. Click **Configuration**. The broker's **Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then either provide its value in the format **name=value**, or remove an existing variable.
7. Click **Save** when you finish adding (or deleting) the variables and their values.

## Editing a WebSpeed Transaction Server configuration

You can edit a WebSpeed Transaction Server configuration.



### To edit a WebSpeed Transaction Server configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **WebSpeed** category, and select a broker. The details page for the broker appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **WebSpeed Configuration** page opens.
5. Click **Edit**. Make the changes you want to the properties. (For details about the properties, see [Table 7–6](#) through [Table 7–18](#).)
6. Click **Save**. Any changes you make are also saved in the `ubroker.properties` file.

## Creating a WebSpeed broker

OpenEdge Management and OpenEdge Explorer provide a sample WebSpeed broker instance (wsbroker1) with a default configuration. Each new instance you create uses the same default configuration, which you can edit.



### To create a WebSpeed broker instance:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **WebSpeed**. The **WebSpeed Configuration** page appears.
4. Type the name of the new WebSpeed broker instance in the field provided.

---

**Note:** The WebSpeed broker instance name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured WebSpeed broker instance names.

---

5. Click **Save**. The **WebSpeed Configuration** page appears, allowing you to configure the WebSpeed broker's properties.

Each new WebSpeed broker instance you create uses the default configuration. However, several properties, such as port numbers, must have unique values for each WebSpeed broker in order for the instance to operate properly.

## Starting and stopping a WebSpeed broker

From the management console, you can start either a local WebSpeed broker or a WebSpeed broker running under a remote AdminServer on another machine. You must first connect to the AdminServer that manages the WebSpeed broker you want to start.

You can also choose to start a WebSpeed broker automatically when the AdminServer starts.

---

**Note:** Prior to starting a WebSpeed broker using SonicMQ ServerConnect, you must first configure the WebSpeed broker as SonicMQ ServerConnect enabled.

---



### To start a WebSpeed broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker instance you want to start. The **Details** page for the instance appears in the detail frame.
3. Click **Control**. That instance's **Control** page appears.
4. Click **Start WebSpeed**.

## Starting a WebSpeed broker automatically

You can set a WebSpeed broker to start automatically when the AdminServer starts.



### To start a WebSpeed broker automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker instance you want to start automatically. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The **WebSpeed Configuration** page appears.
4. Click **Edit**.
5. Select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.

## Stopping a WebSpeed broker

From the management console, you can directly stop a WebSpeed broker on the machine where it runs.



### To stop a WebSpeed broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker instance you want to stop. The **Details** page for that instance appears in the detail frame.
3. Click **Broker Control**. The **WebSpeed Control** page appears.
4. Click **Stop WebSpeed**.

## Adding or trimming WebSpeed agents

Using OpenEdge Management or OpenEdge Explorer, you can start additional WebSpeed agents, up to the number specified in the Maximum Agents property setting.

You can also trim running WebSpeed agents, down to the Minimum Agents property setting.

### More information about trimming WebSpeed Agents

To better understand the use of the Agent Auto-trim feature, consider a scenario involving a broker serving a client/server application running in stateless mode. Assume that the following property values are set:

- Minimum Agent/Server Instances: 10
- Initial Agent/Server Instances to Start: 10
- Maximum Agent/Server Instances: 50
- Agent/Server Auto-trim Interval: 1800

When the broker starts, it starts 10 agents. This also marks the start of the first Agent/Server Auto-trim Interval. Assume that the broker is idle for 30 minutes (it is just before people arrive for work, for instance). At the end of the 30-minute interval, the highest number of agents that were busy at the same time would be zero. (Since there were no requests made of any of the agents, none of the agents was busy.) However, since zero is less than the Minimum Agent/Servers Instances value of 10, no automatic trimming takes place. This marks the start of the second interval.

As the second interval starts, employees begin their workday. There is a sudden demand for agents; in fact, the broker receives so many requests that it must start 20 additional agents. Assume that the high volume of workload, and demand on the agents, continues through the second interval. At the interval's end, the maximum number of busy agents is 30, which is the number the broker scaled up to in response to high demand. Thirty agents are running, and the maximum busy level is 30. No trimming occurs.

During the next interval, the request load diminishes somewhat. At the end of the interval, only 20 agents are concurrently busy. The maximum number of busy agents for the interval remains at 30, since 30 were busy at the start of the interval. Thirty agents are running, and the maximum busy level is 30. No trimming occurs.

Assume that all employees have to attend a company-wide meeting that begins midway through the interval. The highest number of agents busy at that same time was 20. At the end of the interval, there are 30 agents running, with zero agents currently busy. Since the maximum busy level is 20 (at the start of the interval), and the number of running agents is 30 (all are available since everyone is at the company meeting), the broker attempts to trim back the number of agents to 20.

Assume that the company meeting ran for another half-hour, and the entire next interval elapsed without any requests. The maximum busy agents is zero, with 20 currently running. Since the Minimum Agent/Server Instances property value is set to 10, the broker attempts to trim back to 10 agents.

Keep in mind that the lower you set the Agent/Server Auto-trim Interval value, the more quickly the broker reacts to a drop in the workload. Under a reasonably steady work load, the broker should trim the number of running agents to the number needed. However, if the workload fluctuates a great deal and the Agent/Server Auto-trim Interval is set too low, the broker may trim agents that it must then restart. As stated earlier, repetitive trimming and restarting causes the system to waste resources on unnecessary process management.

**To add or trim WebSpeed agents:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker for which you want to add or trim agents. The **Details** page for that broker appears in the detail frame.
3. If the WebSpeed broker is not started, click **Broker Control**. Then click **Start WebSpeed**.
4. When the WebSpeed broker is running, return to the **Details** page.
5. Click **Agent Pool Control**.
6. Review the **Agent pool initial configuration**, the **Agents state**, and the **Agents pool summary** information.
7. To add or trim agents, choose the corresponding option in the **Add/Trim** dropdown menu, and type the number of agents in the box provided.
8. Click **Submit**.

## Deleting a WebSpeed broker

You can delete any WebSpeed broker instance listed in the management console as long as the instance is not running. If the instance is running, you must stop it and then delete it.

**To delete a WebSpeed broker instance:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the WebSpeed broker instance you want to delete. The **Details** page for that instance appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## For additional WebSpeed information

You can find additional information about using OpenEdge Management and/or OpenEdge Explorer with WebSpeed in *OpenEdge Management: Servers, DataServers, Messengers, and Adapters*. See the document for details, including specifics about the following:

- Working with the WebSpeed log file monitor and viewer
- Killing a WebSpeed process



---

## Configuring SonicMQ Adapters

---

You can work with SonicMQ Adapter instances in OpenEdge Management and OpenEdge Explorer, as described in the following sections:

- [SonicMQ Adapter Broker configuration and administration](#)
- [Working with the SonicMQ Adapter](#)
- [Configuring a SonicMQ Adapter](#)
- [Editing a SonicMQ Adapter configuration](#)
- [Creating a SonicMQ Adapter broker](#)
- [Deleting a SonicMQ Adapter broker](#)
- [Starting a SonicMQ Adapter](#)
- [Stopping a SonicMQ Adapter](#)
- [Viewing the status of a SonicMQ Adapter broker](#)
- [Viewing the SonicMQ Adapter log file](#)

## SonicMQ Adapter Broker configuration and administration

The OpenEdge SonicMQ® Adapter allows OpenEdge applications to communicate via JMS Messaging through SonicMQ. The SonicMQ Adapter consists of three connection modes:

- **OpenEdge Adapter for SonicMQ ClientConnect (ClientConnect)** — ClientConnect is for OpenEdge clients and runs transparently as a background process in conjunction with an ABL client or OpenEdge Application Server agent process, with a single adapter process per client process. The application running on the ABL client handles messaging control. ClientConnect takes little or no configuration.
- **OpenEdge Adapter for SonicMQ ServerConnect (ServerConnect)** — ServerConnect is for OpenEdge Application Servers (WebSpeed and AppServer). With this configuration there is a single adapter process per AppServer process, allowing multiple Application Server agents to connect to this single adapter process. ServerConnect is configured at the server.
- **OpenEdge Adapter for SonicMQ BrokerConnect (BrokerConnect)** — BrokerConnect is for ABL client applications. It runs as a separate server process to handle OpenEdge client requests. BrokerConnect is a Unified Broker product, part of the AppServer administration framework. You can use OpenEdge Management or OpenEdge Explorer and the command-line tools `adaptconfig` and `adaptman` on all supported platforms, to manage BrokerConnect. You can also edit its properties in the `ubroker.properties` file. No configuration is required within the SonicMQ environment.

---

**Note:** For BrokerConnect, the OpenEdge installation program creates one instance of the OpenEdge SonicMQ Adapter. In most circumstances, this single adapter instance is sufficient. Although it is possible to create additional instances, there is usually no reason to run multiple OpenEdge SonicMQ Adapter instances on the same host. Each instance of BrokerConnect runs as a broker process. This process is multi-threaded, with one thread for each active client application; it can connect to any SonicMQ Broker. With the SonicMQ Adapter, you can write ABL applications that access JMS messaging using SonicMQ.

---

You can use OpenEdge Management or OpenEdge Explorer to configure a new instance of BrokerConnect and administer the configuration of existing instances.

## Working with the SonicMQ Adapter

To work with a SonicMQ Adapter, select the instance. For information about the tasks you can perform, see the following sections:

- [Configuring a SonicMQ Adapter](#)
- [Starting a SonicMQ Adapter](#)
- [Creating a SonicMQ Adapter broker](#)
- [Starting a SonicMQ Adapter](#)
- [Stopping a SonicMQ Adapter](#)
- [Viewing the status of a SonicMQ Adapter broker](#)
- [Deleting a SonicMQ Adapter broker](#)
- [Viewing the SonicMQ Adapter log file](#)

## Configuring a SonicMQ Adapter

From the management console, you can view or edit the following default configuration properties of a SonicMQ Adapter:

- Broker
- Server
- SSL

You can also create a new SonicMQ Adapter instance, set environment variables, view an instance's status, delete an instance you no longer need, or view the instance's log file.

Each of these tasks is described in the following sections.

### Viewing or modifying SonicMQ Adapter broker properties

You can view or modify general, owner information, controlling NameServer, appservice name list, logging setting, and advanced features properties for a SonicMQ Adapter instance.



**To view or modify SonicMQ Adapter broker properties:**

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **SonicMQ Adapter** category, and select a SonicMQ Adapter instance. The **Details** page for that instance appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **SonicMQ Adapter Configuration** page opens with the **Broker** properties in focus.

From this page, you can do the following:

- View the SonicMQ Adapter instance's broker properties as described in [Table 8–1](#), [Table 8–2](#), [Table 8–3](#), [Table 8–4](#), [Table 8–5](#), and [Table 8–6](#). All of these tables follow this procedure.
- Click **Edit** to modify the properties. For details, see the [“Starting a SonicMQ Adapter”](#) section on page 8–16.

**Table 8–1: SonicMQ Adapter Broker General properties**

(1 of 2)

Property	Description
Working directory	The SonicMQ Adapter working directory.

**Table 8–1: SonicMQ Adapter Broker General properties** (2 of 2)

Property	Description
<b>Port number</b>	The number of the TCP/IP port that the SonicMQ Adapter broker listens to. Each broker defined in the <code>ubroker.properties</code> file must have a unique port number.
<b>Auto start</b>	A checkbox that you select if you want the SonicMQ Adapter broker to start automatically when the AdminServer starts.

**Table 8–2: SonicMQ Adapter Broker Owner Information properties**

Property	Description
<b>Group name</b>	(UNIX only) The name of the group in UNIX.
<b>Username</b>	The ID of a user account that has system-administration rights.
<b>Password</b>	The user account's password. Note that this field is labeled Group Name if the AdminServer runs on a non-Windows system.

**Table 8–3: SonicMQ Adapter Broker Controlling NameServer properties**  
(1 of 2)

Property	Description
<b>Register with NameServer</b>	A checkbox that indicates whether or not to register the SonicMQ Adapter with a controlling NameServer.
<b>Controlling NameServer</b>	The name of the NameServer with which this SonicMQ Adapter registers. (If you did not select to register the broker with a NameServer, the field is dimmed and unavailable.)

**Table 8–3: SonicMQ Adapter Broker Controlling NameServer properties**  
(2 of 2)

Property	Description
<b>Registration mode</b>	<p>How the broker specifies its hostname if it is registering with a controlling NameServer. This hostname information is passed onto a client application when it attempts to connect to an application service that the broker supports. The choices are:</p> <ul style="list-style-type: none"> <li>• <b>Register-IP Address</b> — (Default) This setting is the most efficient mechanism, and can be used in most cases. It registers with the IP address of the machine where the broker is located.</li> <li>• <b>Register-LocalHost</b> — The broker registers with the hostname of the machine that it runs on. Use this setting when the broker runs on a machine with a single hostname and more than one IP address.</li> <li>• <b>Register-HostName</b> — The broker registers with the values specified in the hostName property. Use this setting when your clients need a fully qualified host name to connect to a broker in a different DNS domain.</li> </ul>
<b>Use host name</b>	<p>Specifies the hostname value to be used if the <b>Registration mode</b> property is set to <b>Register-HostName</b>. In all other cases, this property is ignored.</p>

**Table 8–4: SonicMQ Adapter Broker AppServer Name List properties**

Property	Description
<b>Application service names</b>	<p>The application services that this SonicMQ Adapter broker registers with the controlling NameServer. A SonicMQ Adapter client connecting to the broker must specify one of these service names. You can add or delete service names from the list.</p>

**Table 8–5: SonicMQ Adapter Broker Logging Setting properties**

Property	Description
<b>Broker log filename</b>	The broker log file for this SonicMQ Adapter broker instance.
<b>Broker logging level</b>	<p>The amount of information written to the broker log. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> —Logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> —Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> —Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
<b>Append to broker log file</b>	A checkbox that you select if you want the SonicMQ Adapter broker to look for the log file each time it starts up. If the log file is present, the new information is added to the end of the existing file. If there is no log file, a new file is created.
<b>Broker logging entry types</b>	A single entry or comma delimited list of logging entry types.
<b>Broker log file threshold size</b>	A limit on how large the log file can get (in bytes). When the log is full, the broker creates a new log file with a sequence number (1 to 999,999). Possible values: 500000 - 2147483647.
<b>Maximum number of broker log files</b>	A limit on how many broker log files are kept on the system when there is a threshold limit. The value represents the number of log files including the current one. Includes any rolled over log files that already exist when process started.

**Table 8–6: SonicMQ Adapter Broker Advanced Features properties**

Property	Description
<b>Maximum client instances</b>	The maximum number of client connections the SonicMQ Adapter broker can support concurrently. The default value 512 is high enough to ensure that the number of client connections is unlimited. Realistically, however, the system-level resources required to support an unlimited number of clients might be exhausted before this limit is reached. If you see broker log entries that contain Exception Messages for "out of space" or "OutOfMemory" you might need to reduce the maximum number of client connections to a more reasonable value. Note that lowering this value can cause some client requests to be rejected ("Exceeded Max Clients"). Configuring and starting multiple brokers to handle higher client loads will alleviate the problem.
<b>Priority weight (0-100)</b>	Defines a load-balancing priority for this broker relative to other brokers.
<b>Registration retry (in seconds)</b>	The number of seconds that pass between broker "keep-alive" messages. After registering with its controlling NameServer, the SonicMQ Adapter broker periodically sends "keep-alive" messages to the NameServer to let the NameServer know that the broker is still active. (The default is 30.)
<b>TCP/IP version</b>	The TCP/IP version.

## Viewing or modifying SonicMQ Adapter server properties

You can view or modify general, logging setting, or pool range properties for a SonicMQ Adapter server instance.



### To view or modify SonicMQ Adapter server properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **SonicMQ Adapter** category, and select a SonicMQ Adapter instance. The **Details** page for that instance appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **SonicMQ Adapter Configuration** page opens with the **Broker** properties in focus.



5. Click the **Server** tab.

From this page, you can do the following:

- View a read-only display of the SonicMQ Adapter instance's server properties as described in [Table 8–7](#), [Table 8–8](#), and [Table 8–9](#). All of these tables follow this procedure.
- Click **Edit** to modify the broker properties. For details, see the “[Starting a SonicMQ Adapter](#)” section on page 8–16.

**Table 8–7: SonicMQ Adapter Server General property**

Property	Description
Server startup parameters	The startup parameters that you want to specify for the SonicMQ server

**Table 8–8: SonicMQ Adapter Server Logging Settings properties**

Property	Description
Server log filename	The server log file for this SonicMQ Adapter broker instance.
Server logging level	<p>The amount of information written to the server log. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>None</b> — Log no entries. This is equivalent to turning logging off.</li> <li>• <b>Error Only</b> — Log OpenEdge error messages. This includes all error messages and is unrelated to the entry types specified. Errors continue to be logged at all higher levels.</li> <li>• <b>Basic</b> — Logging entry types determine the logged information. Each entry type generates at least some output. This is the default.</li> <li>• <b>Verbose</b> — Logging entry types determine the logged information, but it is typically more information than <b>Basic</b>.</li> <li>• <b>Extended</b> — Logging entry types determine the logged information, but it is typically more information than <b>Verbose</b>.</li> </ul>
Append to server log file	A checkbox that you select if you want the SonicMQ Adapter broker to look for the log file each time it starts up. If the log file is present, the new information is added to the end of the existing file. If there is no log file, a new file is created.

**Table 8–9: SonicMQ Adapter Server Pool Range properties**

Property	Description
<b>Initial number of threads to start</b>	The number of threads you want to start initially within a SonicMQ Adapter broker process. The default is 20.
<b>Minimum threads</b>	The minimum number of threads to start. The default is 10.
<b>Maximum threads</b>	The maximum number of threads to start. The default is 30.

## Viewing or modifying SonicMQ Adapter SSL properties

You can view or modify general and advanced features SSL properties for a SonicMQ Adapter instance.



### To view or modify SonicMQ Adapter SSL properties:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, expand the **OpenEdge** category.
3. Expand the **SonicMQ Adapter** category, and select a SonicMQ Adapter instance. The **Details** page for that instance appears in the detail frame.
4. In the **Command and control** section of the page, click **Configuration**. The **SonicMQ Adapter Configuration** page opens with the **Broker** properties in focus.
5. Click the **SSL** tab.

From this page, you can do the following:

- View a read-only display of the SonicMQ Adapter instance's SSL properties as described in [Table 8–10](#) and [Table 8–11](#). These tables follow this procedure.
- Click **Edit** to modify the broker properties, as described in [Table 8–10](#) and [Table 8–11](#). For details, see the “[Editing a SonicMQ Adapter configuration](#)” section on page 8–13.

**Table 8–10: SonicMQ Adapter SSL General properties**

Property	Description
<b>Enable SSL client connections</b>	If this property is checked, specifies that all connections to this SonicMQ Adapter must use SSL tunneling.
<b>Private key/digital certificate alias name</b>	The alias name within the OpenEdge keystore of the private key and digital certificate entry to use to authenticate all connections to this SonicMQ Adapter. If not otherwise selected, the SonicMQ Adapter uses the default_server server certificate alias.
<b>Password to access the key/certificate</b>	<p>The password to use for accessing the private key and digital certificate. You must specify a password when you specify the select an OpenEdge keystore alias name.</p> <p><b>Note:</b> Your password is encrypted in the <code>ubroker.properties</code> file. If you set the password in <code>ubroker.properties</code> manually, you must specify the password as an encrypted value that you can obtain using the <code>genpassword</code> utility located in the <code>bin</code> directory of your OpenEdge installation.</p> <p>If you use the default_server server certificate, it also has a default password that you do not need to specify.</p>
<b>Password confirm</b>	<p>Confirmation of the username account password. This field appears only if you are editing the configuration.</p> <p>Note that this field does not appear if the AdminServer is running on a non-Windows host machine.</p>

**Table 8–11: SonicMQ Adapter SSL Advanced Features properties**

Property	Description
<b>Disable SSL session caching</b>	If this property is checked, caching for the SSL client session is disabled.
<b>SSL session cache timeout</b>	The length of time, in seconds, that an SSL client session is held in the session cache, during which an SSL client can resume its session. The default is 180.

## Setting or deleting SonicMQ Adapter environment variables

You can set or delete SonicMQ Adapter environment variables.



### To set or delete SonicMQ Adapter environment variables:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter broker whose environment variables you want to set. The **Details** page for that broker appears in the detail frame.
3. Click **Configuration**. The broker's **SonicMQ Adapter Configuration** page appears.
4. Click **Edit**.
5. Click **Environment Variables**.
6. Enter each variable name and then provide its value in the format **name=value**.
7. Click **Save** when you finish adding (or deleting) the variables and their values.

## Enabling or disabling a SonicMQ Adapter broker

You can enable or disable a SonicMQ Adapter broker.



### To enable or disable a SonicMQ Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter broker you want to enable or disable. The **Details** page for that broker appears in the detail frame.
3. Click **Control**. The broker's **Control** page appears.
4. Click **Edit**, and either select the **Enabled** check box to enable the broker or clear the check box to disable the broker.
5. Click **Save**.

## Editing a SonicMQ Adapter configuration

You can edit a SonicMQ Adapter configuration in the management console.



### To edit a SonicMQ Adapter configuration:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter whose configuration you want to edit. The **Details** page for that broker appears in the detail frame.
3. Click **Configuration**. The **Configuration** page appears.
4. Click **Edit**. (For details about the properties, see [Table 8–1](#) through [Table 8–11](#).)
5. Make the configuration changes, and then click **Save**.

## Creating a SonicMQ Adapter broker

OpenEdge Management and OpenEdge Explorer provide a sample SonicMQ Adapter broker instance (SonicMQ1) and use it as a default. A new instance you create also uses the sample configuration, which you can modify.



### To create a SonicMQ Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the **Details** frame, click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **SonicMQ Adapter**. The **SonicMQ Configuration** page appears.
4. Type the name of the new broker instance in the field provided.

---

**Note:** The SonicMQ Adapter broker instance name is case sensitive and can include any character except a period (.) or square brackets ([ ]). The name must be unique among all configured SonicMQ Adapter broker instance names.

---

5. Click **Save**. The **SonicMQ Adapter Configuration** page appears, allowing you to configure the instance's properties.

You can now set the configuration properties for this instance. When a new instance is created, it has a copy of the default configuration. However, several properties, such as port numbers, must have unique values for each SonicMQ Adapter broker in order for the instance to operate properly.

## Deleting a SonicMQ Adapter broker

You can delete an inactive SonicMQ Adapter broker instance from the management console. If the instance is running, you cannot delete it. You must stop the instance before you can delete it.



### To delete a SonicMQ Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter broker instance you want to delete. The **Details** page for that instance appears in the detail frame.
3. Click **Delete**, and then click **OK** to confirm the deletion.

## Starting a SonicMQ Adapter

From the management console, you can start a SonicMQ Adapter instance. You can also choose to start the instance automatically when the AdminServer starts.



### To start a SonicMQ Adapter:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter instance you want to start. The **Details** page for the instance appears in the detail frame.
3. Click **Control**. That instance's **SonicMQ Adapter Control** page appears.
4. Click **Start SonicMQ Adapter**.

### Starting a SonicMQ Adapter automatically

You can set a SonicMQ Adapter to start automatically when the AdminServer starts.



### To start a SonicMQ Adapter automatically:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter instance you want to start automatically. The **Details** page for that instance appears in the detail frame.
3. Click **Configuration**. The **SonicMQ Adapter Configuration** page appears.
4. Click **Edit**.
5. On the **Broker** tab, select the **Auto start** option by clicking in the checkbox.
6. Click **Save**.



## Stopping a SonicMQ Adapter

You can stop a SonicMQ Adapter broker from OpenEdge Explorer.



**To stop a SonicMQ Adapter broker:**

1. Click **Resources** in the OpenEdge Explorer console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the SonicMQ Adapter broker you want to stop. The **Details** page for that instance appears in the detail frame.
3. Click **Control**. The **SonicMQ Adapter Control** page appears.
4. Click **Stop SonicMQ Adapter**.

## Viewing the status of a SonicMQ Adapter broker

You can view the status of an active SonicMQ Adapter broker in the management console.



### To view the status of a SonicMQ Adapter broker:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired SonicMQ Adapter instance. The **Details** page for the instance appears in the detail frame.
3. Click **Status**. A Status summary appears and provides the following information:
  - **Host** — The name of the host machine.
  - **Broker Name** — The name of the broker whose status you are viewing.
  - **Operating Mode** — For SonicMQ Adapter brokers, the operating mode is stateless.
  - **Broker Status** — The current state of the broker.
  - **Broker Port** — The TCP/IP port number that the broker listens to.
  - **Broker PID** — The process ID number of the broker.
  - **Active Servers** — The number of active SonicMQ Adapter brokers.
  - **Busy Servers** — The number of SonicMQ Adapter brokers that are currently processing requests.
  - **Locked Servers** — The number of SonicMQ Adapter brokers that have been locked.
  - **Available Servers** — The number of available SonicMQ Adapter brokers.
  - **Active Clients (now, peak)** — The number of client connections at the current point, and the highest number of connections since the broker started.
  - **Client Queue Depth (cur, max)** — The number of clients waiting for agents to become available to service their request. The current value (**cur**) represents the number of waiting clients at the moment the status is displayed, and the maximum value (**max**) represents the largest number of clients waiting concurrently since the broker was started.
  - **Total Requests** — The total number of requests from clients since the broker started.
  - **Rq Wait (max, avg)** — The maximum and average time the client had to wait for an available broker.
  - **Rq Duration (max, avg)** — The maximum and average time the request required.

For each server, the following details are also provided:

- **Svr#** — The process ID number of the broker process
- **State** — The current state of the broker process
- **Port** — The TCP/IP port number used by the broker
- **nRq** — The total number of messages sent to the broker since it started
- **nRcvd** — The total number of messages received by the broker since it started
- **nSent** — The total number of requests sent by the broker
- **Started** — The time stamp of when the broker process started
- **Last Change** — The time stamp of when the server process last changed state

## Viewing the SonicMQ Adapter log file

You can view the log file for a SonicMQ Adapter by using the log file viewer. The log file viewer allows you to examine the log file through an HTML interface.



### To access the log file viewer:

1. Click **Resources** in the management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired broker. The **Details** page for that broker appears in the detail frame.
3. Click **Log File Viewer**.

You can use the Log File Viewer in the following ways:

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

---

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

---

- Click **Reload** after changing the values in either the **Show** field or the **Overlap** field. If you do not reload, the viewer continues to display 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, regardless of 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 **Previous** 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**.

- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.
- OpenEdge Management and OpenEdge Explorer consider a viewer that has been inactive for more than four hours "stale" and at that point release ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management and OpenEdge Explorer automatically reload 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 and OpenEdge Explorer consider a viewer that has been inactive for forty-eight hours "dead." Once a viewer dies, OpenEdge Management and OpenEdge Explorer release all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.



---

## Reference to OpenEdge Web Service Properties

---

You can set the properties described in this appendix to affect execution of a deployed Web service. The properties that are available for you to set depend on the session model of the Web service (session managed or session free). You can verify the session model of a Web service by viewing its status.

The properties described in this appendix apply to an OpenEdge Web service whether you deploy it as a Web service using an instance of the Web Services Adapter (WSA) in the OpenEdge environment or you deploy it as an OpenEdge service using the OpenEdge Adapter for Sonic ESB to install it in the Sonic Enterprise Service Bus (ESB) environment. In the Sonic ESB environment an OpenEdge service can function either as a Web service (as in the OpenEdge environment) or in integration with other processes on the Enterprise Service Bus. In all cases, these properties have the same effect on service functionality.

This appendix contains the following sections:

- [Web services overview](#)
- [Alphabetical reference](#)

## Web services overview

You can set default values for Web service properties on a WSA instance using OpenEdge Management or OpenEdge Explorer. These values then apply to every Web service that you deploy to that WSA instance. Once deployed, you can review and change these values for each individual Web service.

You can set the same properties for an OpenEdge service in the Sonic ESB environment, using the custom resource editor for OpenEdge services accessible from the Sonic ESB Explorer.

In the following sections, *service* refers to either an OpenEdge Web service or an OpenEdge service, and *adapter* refers to either the WSA Adapter or the OpenEdge Adapter for Sonic ESB.

## Summary of properties

Table A–1 lists a summary of all the properties that you can set for a service, depending on its session model (free, managed, or both).

**Table A–1: Service properties** (1 of 3)

Property	Session model	Short description
<code>appServiceHost</code>	Both	Host name for the NameServer or AppServer that supports an application service
<code>appServiceName</code>	Both	The name of an application service supported by the specified NameServer or AppServer and that supports all of the operations that define the service
<code>appServicePort</code>	Both	The port number to access the application service bound by the service
<code>appServiceProtocol</code>	Both	The protocol that the adapter instance uses to access the host for the specified NameServer or AppServer
<code>connectionLifetime</code>	Free	The maximum lifetime (in seconds) of AppServer connections in the connection pool for this service
<code>idleSessionTimeout</code>	Free	The duration (in seconds) between attempts by the adapter to shut down extra network connections to the AppServer, based on client demand
<code>initialSessions</code>	Free	The number of network sessions to be created (and shared by all clients) when the connection pool for the service is initialized by the adapter
<code>maxSessions</code>	Free	The maximum number of connected sessions allowed in the service connection pool



**Table A–1: Service properties**

(2 of 3)

Property	Session model	Short description
<code>minIdleConnections</code>	Free	The minimum number of idle, or inactive, AppServer connections to maintain
<code>minSessions</code>	Free	The minimum number of connected sessions that the adapter attempts to maintain in the service connection pool
<code>noHostVerify</code>	Both	Controls whether the service verifies that the host name of the connecting AppServer matches the Common Name specified in the server digital certificate (meaningful only in the context of SSL connections)
<code>noSessionReuse</code>	Both	Controls whether the service requests reuse of the SSL session ID when reconnecting to the same AppServer (meaningful only in the context of SSL connections)
<code>nsClientMaxPort</code>	Both	The maximum value for the adapter to specify for the port number used to communicate with a NameServer
<code>nsClientMinPort</code>	Both	The minimum value for the adapter to specify for the port number used to communicate with a NameServer
<code>nsClientPicklistExpiration</code>	Free	The maximum duration (in seconds) that the adapter retains a list of AppServer options (pick list) for an idle application service
<code>nsClientPicklistSize</code>	Free	The number of available AppServer options (the broker pick list) that the adapter requests from a NameServer each time it looks up a given application service name
<code>nsClientPortRetry</code>	Both	The maximum number of requests that the adapter makes for a valid local port number when attempting to communicate with a NameServer
<code>nsClientPortRetryInterval</code>	Both	The interval (in milliseconds) that the adapter waits between requests to get a valid port number when attempting to communicate with a NameServer
<code>requestWaitTimeout</code>	Free	Determines how the adapter handles requests when the service connection pool becomes full
<code>serviceFaultLevel</code>	Both	The amount of information returned to the client for a SOAP Fault
<code>serviceLoggingLevel</code>	Both	The amount and type of information written to the service log by the adapter for each log entry

**Table A–1: Service properties**

(3 of 3)

Property	Session model	Short description
<code>stale04GLObjectTimeout</code>	Both	The maximum duration (in seconds) that a service object can be idle before it is released
<code>waitIfBusy</code>	Managed	Determines how to handle client requests to a service that is busy processing a prior request

---

**Note:** A *connection pool* is a cache that the adapter manages for each session-free service that relies on a controlling NameServer to access the AppServer that provides the corresponding application service. This cache maintains one or more AppServer connections for the application service, which are made available by the NameServer and otherwise managed through these properties. For more information on connection pools, see *OpenEdge Application Server: Administration* on the Product Software Developers Network (PSDN).

---

## Setting service properties

You must disable a service before you can set most properties on it. You can verify if the service is disabled by viewing its status. The only properties you can set for an enabled service are:

- `serviceFaultLevel`
- `serviceLoggingLevel`

If you set other properties while the service is enabled, the property value changes take effect only after you disable, then enable the service again.

## Alphabetical reference

This section contains an alphabetical reference to the properties that you can set for a service, including the applicable session model, a complete description of usage, and the default value for each property.

### appServiceHost

**Session model:** Managed or Free

Host name for the NameServer or AppServer that supports an application service with the name specified by the [appServiceName](#) property.

**Installation default:** localhost

### appServiceName

**Session model:** Managed or Free

The name of an application service supported by the NameServer or AppServer specified by the [appServiceHost](#) property, and that supports all operations that define the service.

**Installation default:** asbroker1

### appServicePort

**Session model:** Managed or Free

The UDP (NameServer) or TCP (AppServer) port number to access the application service supported by the host specified by the [appServiceHost](#) property.

**Installation default:** 5162

### appServiceProtocol

**Session model:** Managed or Free

The protocol that the WSA Adapter or OpenEdge Adapter for Sonic ESB uses to access the host specified by the [appServiceHost](#) property. For a NameServer host, this is “AppServer”; for an AppServer host, “AppServerDC”.

**Installation default:** AppServer

## connectionLifetime

**Session model:** Free

The maximum lifetime (in seconds) of AppServer connections in the connection pool for this service. A value of 0 indicates that the lifetime of these connections is unlimited, unless they are disconnected according to the requirements of other property settings, such as [idleSessionTimeout](#). A positive value indicates that the WSA Adapter or OpenEdge Adapter for Sonic ESB maintains any AppServer connections for this service for the specified number of seconds. Thus, when the [idleSessionTimeout](#) interval expires, the adapter does any necessary trimming of connections in the connection pool beginning with those whose [connectionLifetime](#) interval has expired. However, the adapter maintains the connections for all services whose [connectionLifetime](#) interval has not yet expired regardless of other property settings.

**Installation default:** 0

## idleSessionTimeout

**Session model:** Free only

The duration (in seconds) between attempts by the WSA Adapter or OpenEdge Adapter for Sonic ESB to shut down extra network connections to the AppServer, based on client demand. The adapter does this by monitoring the maximum number of sessions needed since the last time-out, then disconnecting any connections in excess of that number.

A value of 0 indicates that the adapter will never disconnect idle sessions unless the [connectionLifetime](#) interval has expired.

**Installation default:** 0

## initialSessions

**Session model:** Free only

The number of network sessions to be created (and shared by all clients) when the connection pool for the service is initialized by the WSA Adapter or OpenEdge Adapter for Sonic ESB. This value must be between the value of the [minSessions](#) property and the [maxSessions](#) property, inclusive, unless [maxSessions](#) is set to 0. If [maxSessions](#) is set to 0, the [initialSessions](#) value must only be greater than or equal to [minSessions](#).

**Installation default:** 1

## maxSessions

**Session model:** Free only

The maximum number of connected sessions allowed in the connection pool. Once the number of sessions in the pool reaches the limit specified by [maxSessions](#), the WSA Adapter or OpenEdge Adapter for Sonic ESB creates no additional sessions for this service, and handles all requests for this service according to the [requestWaitTimeout](#) property setting.

A value of 0 indicates that the size of the connection pool is unlimited.

**Installation default:** 0

## minIdleConnections

**Session model:** Free only

The minimum number of idle, or inactive, AppServer connections to maintain. Given sufficient AppServer resources, this setting allows the WSA Adapter or OpenEdge Adapter for Sonic ESB to always have a free connection available when a new request arrives, so the request does not have to wait for the adapter to locate and establish a new connection to service it. When the value of this property is greater than zero, the adapter uses its free time to pre-allocate network connections for the service connection pool.

**Installation default:** 0

## minSessions

**Session model:** Free only

The minimum number of connected sessions allowed in the connection pool. The WSA Adapter or OpenEdge Adapter for Sonic ESB attempts to keep at least this many sessions connected to the application service (AppServer).

**Installation default:** 1

## noHostVerify

**Session model:** Both

If set to 1, turns off host verification for an SSL Web service connection (specified by the [appServiceProtocol](#) property). If cleared, the WSA compares the host name of the connecting AppServer with the Common Name specified in the server digital certificate, and raises a Web service error if they do not match. With this parameter specified, the Web service never raises the error.

**Installation default:** 0

## noSessionReuse

**Session model:** Both

If set to 1, the Web service connection does not reuse the SSL session ID when reconnecting to the same AppServer for an SSL Web service connection (specified by the [appServiceProtocol](#) property).

**Installation default:** 0

## nsClientMaxPort

**Session model:** Managed or Free

The maximum value for the UDP port number that the WSA Adapter or OpenEdge Adapter for Sonic ESB specifies when communicating with the NameServer. This value must be greater than or equal to the value of the [nsClientMinPort](#) property.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 0

## nsClientMinPort

**Session model:** Managed or Free

The minimum value for the WSA Adapter or OpenEdge Adapter for Sonic ESB to specify for the UDP port number when communicating with the NameServer. The value must be less than or equal to the value of the [nsClientMaxPort](#) property.

If this value is 0, the adapter chooses the NameServer client port number randomly.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 0

## nsClientPicklistExpiration

**Session model:** Free only

The maximum duration (in seconds) that the WSA Adapter or OpenEdge Adapter for Sonic ESB retains a list of AppServer options (pick list) for an idle application service.

A value of 0 indicates that the pick list never expires.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 0

## nsClientPicklistSize

**Session model:** Free only

The number of available AppServer options (the broker pick list) that the WSA Adapter or OpenEdge Adapter for Sonic ESB requests from the NameServer each time it looks up a given application service name.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 1

## nsClientPortRetry

**Session model:** Managed or Free

The maximum number of requests that the WSA Adapter or OpenEdge Adapter for Sonic ESB makes for a valid local UDP port number when attempting to communicate with the NameServer on behalf of the service.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 3

## nsClientPortRetryInterval

**Session model:** Managed or Free

The interval (in milliseconds) that the WSA Adapter or OpenEdge Adapter for Sonic ESB waits between requests to get a valid UDP port number when attempting to communicate with the NameServer on behalf of the service.

This property applies only to services that use a NameServer to access application services (an AppServer).

**Installation default:** 200

## requestWaitTimeout

**Session model:** Free

Determines how the WSA Adapter or OpenEdge Adapter for Sonic ESB handles requests when the connection pool becomes full. (The connection pool is full when the number of active sessions equals the value of [maxSessions](#), and all sessions are currently running requests.)

The adapter handles such requests according to the value of `requestWaitTimeout`, as shown in [Table A-2](#).

**Table A-2: requestWaitTimeout property values**

If the value is . . .	The WSA Adapter or OpenEdge Adapter for Sonic ESB. . .
-1	Queues the request indefinitely until an AppServer session becomes available.
0	Rejects the request and returns an error message to the client indicating that there are too many concurrent requests.
> 0	Queues the request for the maximum number of seconds specified by the value until an AppServer session becomes available. If no session becomes available in that time, the adapter returns an error to the client.

**Installation default:** 0

## serviceFaultLevel

**Session model:** Managed or Free

The amount (level) of information returned to the client for a SOAP Fault as determined by an integer value. A level of 2 returns basic information in the <FaultCode> and <FaultString> elements for each SOAP Fault message, which is suitable for normal production environments. A level of 3 returns more detailed information that is suitable for development environments. Other values provide varying levels of diagnostic information, and are reserved for use by Progress Technical Support and Engineering.

**Installation default:** 2

## serviceLoggingLevel

**Session model:** Managed or Free

The amount and type of information written for each service log entry by the WSA Adapter or OpenEdge Adapter for Sonic ESB to the adapter log file, cumulatively determined by the integer values from 1 to 4, as shown in [Table A-3](#).

**Table A-3: serviceLoggingLevel property values**

If the value is . . .	The WSA Adapter or OpenEdge Adapter for Sonic ESB logs . . .
1	Errors internally detected by the adapter only (errors that result in SOAP Faults not included)
2	Additional errors that result in SOAP Faults returned to the client
3	Additional debugging information that is helpful to the deployer
4	Additional debugging information that is helpful to Progress support services and engineering

**Installation default:** 2

## staleO4GLObjectTimeout

**Session model:** Managed or Free

The maximum duration (in seconds) that a service object (AppObject, SubAppObject, or ProcObject) can be idle before it is released.

As part of managing certain service objects with OpenEdge, clients explicitly create them using factory methods before invoking other methods on them. When the client no longer requires the object, it has the responsibility to release the object from the service run-time context. However, if this time-out expires before the client releases the object, the WSA Adapter or OpenEdge Adapter for Sonic ESB assumes that the client application no longer requires access to the object, and deletes it from the service run-time context automatically. In effect, the adapter uses this time-out to provide garbage collection on service objects that client applications stop referencing and fail to release in the specified period of time. Any subsequent attempt by a client to access this object returns an error from the adapter.



A value of 0 for this property specifies that the adapter perform no garbage collection on idle objects, leaving the responsibility for releasing them entirely to client applications.

When all objects that reference the connection pool have been released, the adapter also releases the now unreferenced connection pool as well.

**Installation default:** 0

## **waitIfBusy**

**Session model:** Managed only

An integer value that determines how to handle client requests to a service that is busy processing a prior request. If the value is 1, the WSA Adapter or OpenEdge Adapter for Sonic ESB queues multiple requests for this service and executes them one at a time until the queue is empty. If the value is 0 and the adapter is executing a prior request for the service, each subsequent request for the same service fails until the adapter completes the request it is currently executing.

**Installation default:** 0



---

# Index

---

## A

- Activate property 5–16
- Admin role
  - Web Services Adapter 6–14
- AdminServer 1–2
- After image buffers property 2–10
- After image process property 2–10
- After image stall property 2–10
- After-image Writer 2–22
- Agent maximum port property 2–29
- Agent minimum port property 2–29
- Agent name property 2–28
- Agent Shutdown Action property 2–31
- AIW 2–22
- Append to broker log file property
  - AppServer broker 5–7
  - AppServer messaging 5–20
  - MS SQL Server DataServer 3–29
  - ODBC DataServer 3–5
  - Oracle DataServer 3–17
  - WebSpeed broker 7–12
  - WebSpeed messaging 7–26
- Append to log file property 6–11
- Append to NameServer log file property 4–6
- Append to server log file property
  - AppServer agent 5–13
  - AppServer messaging 5–20
  - WebSpeed agent 7–18
  - WebSpeed messaging 7–26
- Application service names property
  - AppServer 5–6
  - WebSpeed 7–11
- Application URL property 7–22
- AppServer Internet Adapters
  - actions allowed on 5–30
  - configuration overview 5–30
  - creating instances 5–37
  - deleting instances 5–40
  - editing a configuration 5–37
  - enabling and disabling 5–39
  - properties 5–31
    - advanced features 5–37
    - controlling NameServer 5–32
    - general 5–32
    - logging setting 5–34
    - SSL 5–36
  - viewing log files 5–38
- AppServers
  - actions allowed on 5–3
  - additional management 5–29
  - agent properties 5–11
    - advanced features 5–16
    - general 5–12
    - logging setting 5–13
    - pool range 5–16
  - agents
    - adding or trimming 5–23

- broker properties 5–4
  - advanced features 5–10
  - AppService Name List 5–6
  - controlling NameServer 5–5
  - general 5–5
  - logging setting 5–7
  - owner information 5–5
- configuration overview 5–2
- creating instances 5–22
- deleting instances 5–28
- editing a configuration 5–21
- environment variables 5–21
- features 5–2
- messaging properties 5–19
- property storage 5–3
- SSL properties 5–17
  - advanced features 5–18
  - general 5–18
- starting
  - automatically 5–23
  - manually 5–22
- status 5–25
- stopping 5–23
- viewing log files 5–28

appServiceHost service property A–5

appServiceName service property A–5

appServicePort service property A–5

appServiceProtocol service property A–5

APW 2–22

Assigning database default configurations 2–15

Asynchronous Page Writer 2–22

Asynchronous page writers property 2–9

Authorized IP list property 7–7

Auto start property

- AppServer 5–5
- MS SQL Server DataServer 3–28
- NameServer 4–6
- ODBC DataServer 3–4
- OpenEdge database 2–5
- Oracle DataServer 3–16
- WebSpeed Transaction Server 7–10

Auto trim timeout property

- AppServer 5–11
- WebSpeed Transaction Server 7–16

Automatically add After imaging extents property 2–35

## B

Background writers 2–22

Backup method property 2–35

Before image buffers property 2–9

Before image cluster age property 2–10

Before image delay writers property 2–10

Before image process property 2–9

Before image stall property 2–9

Before image threshold property 2–9

Before-image Writer 2–22

Begin After imaging after transition property 2–35

Binary upload max size property 7–22

BIW 2–22

Blocks in database buffers property 2–6

Broker keep alive timeout property 4–5

Broker log file threshold size property

- AppServer 5–9
- MS SQL Server DataServer 3–29
- ODBC DataServer 3–5
- Oracle DataServer 3–17
- WebSpeed Transaction Server 7–14

Broker log filename property

- AppServer 5–7
- MS SQL Server DataServer 3–29
- ODBC DataServer 3–5
- Oracle DataServer 3–17
- WebSpeed Transaction Server 7–12

Broker logging entry types property

- AppServer 5–8
- MS SQL Server DataServer 3–29
- ODBC DataServer 3–5
- Oracle DataServer 3–17
- WebSpeed Transaction Server 7–13

Broker logging level property

- AppServer broker 5–7
- AppServer messaging 5–20
- MS SQL Server DataServer 3–29
- ODBC DataServer 3–5
- Oracle DataServer 3–17
- WebSpeed broker 7–12
- WebSpeed messaging 7–26

## C

Case table name property 2–8

### Changing

#### AppServer

- agent properties 5–11
- broker properties 5–4
- environment variables 5–21
- messaging properties 5–19
- SSL properties 5–17

#### AppServer Internet Adapter properties 5–31

#### MS SQL Server DataServer

- broker properties 3–27
- environment variables 3–32
- server properties 3–31

#### NameServer

- environment variables 4–10
- properties 4–4

#### ODBC DataServer

- broker properties 3–3
- environment variables 3–8
- server properties 3–7

#### OpenEdge database

- basic properties 2–5
- default properties 2–6
- server group properties 2–16

#### Oracle DataServer

- broker properties 3–15
- environment variables 3–19
- server properties 3–18

#### SonicMQ Adapter

- broker properties 8–4
- environment variables 8–12
- server properties 8–8
- SSL properties 8–10

#### Web Services Adapter

- advanced features properties 6–14
- location properties 6–8
- logging setting properties 6–11
- properties 6–8
- proxy server setting properties 6–9
- run-time properties 6–23
- security properties 6–13
- WSDL properties 6–10

#### WebSpeed Messenger

- properties 7–2

#### WebSpeed Transaction Server

- agent properties 7–16
- broker properties 7–10
- environment variables 7–27
- messaging properties 7–25
- SSL properties 7–23

Client Type property 2–17

Cluster enabled property 2–5

Collation table property 2–8

Configuration group 2–2

### Configuring

- AppServer components 5–1
- AppServer Internet Adapters 5–30
- AppServers 5–2
- DataServers 3–1
- MS SQL Server DataServers 3–26
- NameServers 4–1
- ODBC DataServers 3–2
- OpenEdge databases 2–1
- OpenEdge Web Services 6–1
- Oracle DataServers 3–14
- SonicMQ Adapters 8–1
- Web service instances 6–25
- Web Services Adapter instances 6–5
- WebSpeed components 7–1
- WebSpeed Messengers 7–2
- WebSpeed Transaction Servers 7–9

conmgr.properties file 2–2

Connect property 5–17

### Connection Timeout property

- replication
  - agent 2–29
  - control agent 2–32

connectionLifetime service property A–6

### Controlling NameServer property

- AppServer 5–5
- MS SQL Server DataServer 3–28
- ODBC DataServer 3–4
- Oracle DataServer 3–16
- WebSpeed Messenger 7–3
- WebSpeed Transaction Server 7–11

Conversion map property 2–8

CPU statistic 2–24

### Creating

- AppServer instances 5–22
- AppServer Internet Adapter instances 5–37
- MS SQL Server DataServer instances 3–33
- NameServer instances 4–12
- ODBC DataServer instances 3–9
- OpenEdge database
  - configurations 2–14
  - server groups 2–19
- Oracle DataServer instances 3–20
- SonicMQ Adapter instances 8–14
- Web Services Adapter instances 6–5
- WebSpeed Transaction Server instances 7–28

Critical agent property 2–33

**D**

- Data service name list property
  - MS SQL Server DataServer 3–29
  - ODBC DataServer 3–5
  - Oracle DataServer 3–17
- Database group 2–2
- Database name property 2–5
- Database property
  - replication
    - agent 2–28
    - control agent 2–32
    - server 2–30
- Database Role property 2–34
- DataServers 3–1
  - See also* ODBC DataServers, Oracle DataServers, MS SQL Server DataServers
- Deactivate property 5–17
- Debug clients property 6–15
- Debug mode property 7–22
- Default configuration property 2–5
- Default cookie domain property 7–22
- Default cookie path property 7–22
- default.props file 6–17
- Defer Agent Startup property 2–30
- Deleting
  - AppServer instances 5–28
  - AppServer Internet Adapter instances 5–40
  - MS SQL Server DataServer instances 3–33
  - NameServer instances 4–13
  - ODBC DataServer instances 3–9
  - OpenEdge database
    - configurations 2–25
    - references 2–25
    - server groups 2–26
  - Oracle DataServer instances 3–21
  - SonicMQ Adapter instances 8–15
  - Web Services Adapter instances 6–24
  - WebSpeed Transaction Server instances 7–31
- Direct I/O property 2–7

- Disable client verification of SSL host name property 7–6
- Disable SSL session caching property 5–18, 7–24
- Disable SSL session reuse property 7–6
- Disconnect property 5–17

**E**

- Editing
  - AppServer Internet Adapter properties 5–37
  - AppServer properties 5–21
  - MS SQL Server DataServer properties 3–32
  - NameServer properties 4–10
  - ODBC DataServer properties 3–8
  - OpenEdge database
    - default properties 2–13
    - server group properties 2–19
  - Oracle DataServer properties 3–20
  - SonicMQ Adapter properties 8–13
  - Web Services Adapter properties 6–16
  - WebSpeed Messenger properties 7–7
  - WebSpeed Transaction Server properties 7–27
- Enable SSL client connections property
  - AppServer 5–18
  - WebSpeed Messenger 7–6
  - WebSpeed Transaction Server 7–24
- Enable Web Services property 6–13
- Enable WSA administration property 6–13
- Enable WSDL listing retrieval property 6–10
- Enable WSDL retrieval property 6–10
- Environment property 4–8

**F**

- Fault tolerance 4–14
  - connection-level 4–14
  - server-level 4–14
- File upload directory property 7–22
- 4GL debugger enabled property
  - AppServer 5–16
  - WebSpeed Transaction Server 7–23
- Full on-line and off-line backup arguments property 2–36

**G**

- Getting started 1–4
- Group name property
  - AppServer 5–5
  - MS SQL Server DataServer 3–28
  - ODBC DataServer 3–4
  - Oracle DataServer 3–16
  - WebSpeed Transaction Server 7–10

**H**

- Hash table entries property 2–6
- Host name property 4–5
- Host property
  - database server group 2–18
  - replication control agent 2–32
- HTTP error page property 6–15

**I**

- idleSessionTimeout service property A–6
- Incremental backup arguments property 2–36
- Initial number of servers to start property
  - AppServer 5–16
  - WebSpeed Transaction Server 7–21
- initialSessions service property A–6
- Internal administration command property 7–6

**J**

- jvmargs property
  - AppServer 5–11
  - WebSpeed Transaction Server 7–16

**K**

- Keep Alive property 2–30
- Kernel time statistic 2–24

**L**

- Load balancing 4–14
  - priority weight 4–15
- Location property
  - NameServer 4–5
  - Web Services Adapter 6–9
- Location transparency 4–15
- Lock table entries property 2–7
- Log code page property 2–8
- Log file threshold size property
  - NameServer 4–7
  - Web Services Adapter 6–12
  - WebSpeed Messenger 7–5
- Log filename property 6–11
- Log files
  - AppServer 5–28
  - AppServer Internet Adapter 5–38
  - MS SQL Server DataServer 3–36
  - NameServer 4–21
  - ODBC DataServer 3–12
  - OpenEdge database 2–3
  - Oracle DataServer 3–24
  - SonicMQ Adapter 8–20
  - WebSpeed Messenger 7–7
- Log message threshold size property 6–12
- Logging entry types property
  - NameServer 4–7
  - Web Services Adapter 6–12
  - WebSpeed Messenger 7–5
- Logging level property
  - Web Services Adapter 6–11
  - WebSpeed Messenger 7–5
- Login
  - Web server 6–8

**M**

- Managed databases 2–2
- Managing database instances 2–4
- Max servers property 2–7
- Max users property 2–7

Maximum client instances property  
  AppServer 5–10  
  MS SQL Server DataServer 3–30  
  ODBC DataServer 3–6  
  Oracle DataServer 3–18  
  WebSpeed Transaction Server 7–15

Maximum clients per server property 2–18

Maximum dynamic port property 2–18

Maximum message property 2–33

Maximum NameServer client port property 7–4

Maximum number of broker log files property  
  AppServer 5–9  
  MS SQL Server DataServer 3–29  
  ODBC DataServer 3–5  
  Oracle DataServer 3–17  
  WebSpeed Transaction Server 7–14

Maximum number of log files property  
  NameServer 4–7  
  Web Services Adapter 6–12  
  WebSpeed Messenger 7–6

Maximum number of server log files property  
  AppServer 5–16  
  WebSpeed Transaction Server 7–21

Maximum port number property  
  AppServer 5–12  
  MS SQL Server DataServer 3–31  
  ODBC DataServer 3–7  
  Oracle DataServer 3–19  
  WebSpeed Transaction Server 7–17

Maximum servers property  
  AppServer 5–16  
  WebSpeed Transaction Server 7–21

maxSessions service property A–6

Message buffer size property 2–17

Messenger executable filename property 7–3

Minimum clients per server property 2–18

Minimum dynamic port property 2–18

Minimum NameServer client port property 7–4

Minimum port number property  
  AppServer 5–12  
  MS SQL Server DataServer 3–31  
  ODBC DataServer 3–7  
  Oracle DataServer 3–19  
  WebSpeed Transaction Server 7–17

Minimum servers property  
  AppServer 5–16  
  WebSpeed Transaction Server 7–21

minSessions service property A–7

Modifying  
  *See* Changing

Monitored property 2–13

MS SQL Server DataServers  
  actions allowed on 3–26  
  broker properties 3–27  
    advanced features 3–30  
    controlling NameServer 3–28  
    data service 3–29  
    general 3–27  
    logging setting 3–29  
    owner information 3–28  
  configuration overview 3–26  
  creating instances 3–33  
  deleting instances 3–33  
  editing a configuration 3–32  
  environment variables 3–32  
  server properties 3–31  
    general 3–31  
    logging setting 3–32  
  starting  
    automatically 3–34  
    manually 3–34  
  status 3–35  
  stopping 3–34  
  viewing log files 3–36

## N

NameServer logging level property 4–6

NameServers  
  actions allowed on 4–3  
  configuration overview 4–2  
  configuring local instances 4–4  
  configuring remote instances 4–9  
  creating instances 4–12  
  deleting instances 4–13  
  editing a configuration 4–10  
  environment variables  
    setting and deleting 4–10  
  features 4–2  
    fault tolerance 4–14  
    load balancing 4–14  
    location transparency 4–15  
  firewalls and UDP 4–20



- local and remote 4–3
- properties
  - advanced features 4–8
  - environment variables 4–8
  - general 4–5
  - location 4–5
  - logging setting 4–6
- starting
  - automatically 4–16
  - manually 4–16
- status 4–18
- stopping 4–17
- viewing log files 4–21

Nap interval property 2–13

Nap max interval property 2–13

Neighboring NameServers property 4–8

Normal startup arguments property 2–35

nsClientMaxPort service property A–8

nsClientMinPort service property A–8

nsClientPicklistExpiration service property A–8

nsClientPicklistSize service property A–8

nsClientPortRetry service property A–9

nsClientPortRetryInterval service property A–9

Number of servers property 2–16

## O

ODBC DataServers

- actions allowed on 3–2
- broker properties 3–3
  - advanced features 3–6
  - controlling NameServer 3–4
  - data service 3–5
  - general 3–3
  - logging setting 3–5
  - owner information 3–4
- configuration overview 3–2
- creating instances 3–9
- deleting instances 3–9
- editing a configuration 3–8
- environment variables 3–8
- server properties 3–7
  - general 3–7
  - logging setting 3–7
- starting
  - automatically 3–10
  - manually 3–9
- status 3–10

- stopping 3–10
- viewing log files 3–12

OpenEdge databases

- actions allowed on 2–3
- assigning a default configuration 2–15
- background writers 2–22
- basic properties 2–5
- configuration overview 2–2
- configuring replication-enabled 2–27
  - See also* Replication-enabled databases
- conmgr.properties file 2–2
- creating a configuration 2–14
- creating a server group 2–19
- default configuration
  - editing 2–13
- default properties 2–6
- deleting references 2–25
- managed 2–2
- managing instances 2–4
- process details 2–23
- property settings
  - awareness 2–3
  - basic 2–5
  - default 2–6
- scripted 2–2
- server group properties 2–16
  - editing 2–19
- shared memory systems 2–22
- starting
  - automatically 2–21
  - manually 2–20
- status 2–23
- stopping 2–20
- viewing log files 2–3
- watchdog process 2–22

OpenEdge Explorer

- comparing OpenEdge Management 1–4
- introduction 1–1
- overview 1–3

OpenEdge Management

- comparing OpenEdge Explorer 1–4
- introduction 1–1
- overview 1–2

OpenEdge Web Services

- application server 6–2
- configuration overview 6–2
- default property settings 6–17
- default.props file 6–17
- deploying to a WSA 6–18
- enabling or disabling for access 6–25
- exporting from a WSA 6–20
- importing to a WSA 6–20
- listing deployed to a WSA 6–21
- login 6–8
- managing 6–25
- managing with the WSA 6–3

- properties
    - alphabetical reference A–5
    - overview A–1
    - setting A–4
    - summary listing A–2
  - run-time statistics 6–26
  - session models 6–3
  - status of deployed Web service 6–26
  - undeploying 6–30
  - updating 6–27
    - definition and deployment information 6–28
    - property settings 6–29
  - Operating mode property
    - AppServer 5–5
    - MS SQL Server DataServer 3–27
    - ODBC DataServer 3–3
    - Oracle DataServer 3–15
    - WebSpeed Transaction Server 7–10
  - Operator role
    - Web Services Adapter 6–14
  - Oracle DataServers
    - actions allowed on 3–14
    - broker properties 3–15
      - advanced features 3–18
      - controlling NameServer 3–16
      - data service 3–17
      - general 3–15
      - logging setting 3–17
      - owner information 3–16
    - configuration overview 3–14
    - creating instances 3–20
    - deleting instances 3–21
    - editing a configuration 3–20
    - environment variables 3–19
    - server properties 3–18
      - general 3–19
      - logging setting 3–19
    - starting
      - automatically 3–21
      - manually 3–21
    - status 3–22
    - stopping 3–22
    - viewing log files 3–24
  - Other arguments property 2–7
    - replication
      - agent 2–29
      - control agent 2–33
      - server 2–31
- P**
- Page writer max buffers property 2–12
  - Page writer queue delay property 2–12
  - Page writer queue minimum property 2–12
  - Page writer scan delay property 2–12
  - Page writer scan property 2–12
  - Password confirm property
    - AppServer broker 5–5
    - AppServer SSL 5–18
    - WebSpeed broker 7–11
    - WebSpeed SSL 7–24
  - Password property
    - AppServer 5–5
    - MS SQL Server DataServer 3–28
    - ODBC DataServer 3–4
    - Oracle DataServer 3–16
    - WebSpeed Transaction Server 7–11
  - Password to access the key/certificate property
    - AppServer 5–18
    - WebSpeed Transaction Server 7–24
  - Polling delay maximum property
    - replication
      - agent 2–29
      - server 2–31
  - Polling delay minimum property
    - replication
      - agent 2–29
      - server 2–31
  - Port number property
    - AppServer 5–5
    - MS SQL Server DataServer 3–27
    - NameServer 4–5
    - ODBC DataServer 3–3
    - Oracle DataServer 3–15
    - WebSpeed Transaction Server 7–10
  - Port property 2–32
  - Priority weight property
    - AppServer 5–10
    - MS SQL Server DataServer 3–30
    - ODBC DataServer 3–6
    - Oracle DataServer 3–18
    - WebSpeed Transaction Server 7–15
  - Private key/Digital certificate alias name property
    - AppServer 5–18
    - WebSpeed Transaction Server 7–24
  - Process details
    - OpenEdge database 2–23
  - Process time statistic 2–24
  - PROPATH property
    - AppServer 5–12
    - WebSpeed Transaction Server 7–17

- Properties
  - AppServer
    - agent 5–11
    - broker 5–4
    - messaging 5–19
    - SSL 5–17
  - AppServer Internet Adapter 5–31
  - MS SQL Server DataServer
    - broker 3–27
    - server 3–31
  - ODBC DataServer
    - broker 3–3
    - server 3–7
  - OpenEdge database
    - basic 2–5
    - conmgr.properties file 2–2
    - default 2–6
    - server group 2–16
    - setting awareness 2–3
  - Oracle DataServer
    - broker 3–15
    - server 3–18
  - SonicMQ Adapter
    - broker 8–4
    - server 8–8
    - SSL 8–10
  - Web service
    - alphabetical reference A–5
    - overview A–1
    - summary listing A–2
  - Web Services Adapter 6–8
  - WebSpeed Messenger 7–2
  - WebSpeed Transaction Server
    - agent 7–16
    - broker 7–10
    - messaging 7–25
    - SSL 7–23
- Proxy host property 6–9
- Proxy password Confirm property 6–10
- Proxy password property 6–10
- Proxy port property 6–9
- Proxy username property 6–9
- R**
  - Register with NameServer property
    - AppServer 5–5
    - MS SQL Server DataServer 3–28
    - ODBC DataServer 3–4
    - Oracle DataServer 3–16
  - WebSpeed Messenger 7–3
  - WebSpeed Transaction Server 7–11
  - Registration host name property
    - AppServer 5–6
    - MS SQL Server DataServer 3–28
    - ODBC DataServer 3–4
    - Oracle DataServer 3–16
    - WebSpeed Transaction Server 7–11
  - Registration mode property
    - AppServer 5–6
    - MS SQL Server DataServer 3–28
    - ODBC DataServer 3–4
    - Oracle DataServer 3–16
    - WebSpeed Transaction Server 7–11
  - Registration retry property
    - AppServer 5–10
    - MS SQL Server DataServer 3–30
    - ODBC DataServer 3–6
    - Oracle DataServer 3–18
    - WebSpeed Transaction Server 7–15
  - Replication method property 2–33
  - Replication-enabled databases 2–27
    - property settings
      - agent 2–28
      - categories 2–28
      - control agent 2–32
      - rules 2–27
      - server 2–30
      - transition 2–33
  - Reporting interval property 2–17
  - Request timeout property
    - AppServer 5–10
    - WebSpeed Transaction Server 7–15
  - requestWaitTimeout service property A–9
  - Require Web Services authorization
    - property 6–13
  - Require WSA administration authorization
    - property 6–13
  - Require WSDL Retrieval authorization
    - property 6–13
  - Resident size statistic 2–24
  - Responsibility property 2–34
  - Restart after transition property 2–34
  - Roles
    - Web Services Adapter 6–14

**S**

- Schema Lock Action property 2–31
- Scripted databases 2–2
- Semaphore sets property 2–11
- Server application mode property 7–17
- Server code page property 2–8
- Server executable file property
  - AppServer 5–12
  - MS SQL Server DataServer 3–31
  - ODBC DataServer 3–7
  - Oracle DataServer 3–19
  - WebSpeed Transaction Server 7–17
- Server log file name property
  - ODBC DataServer 3–7
- Server log file threshold size property
  - AppServer 5–15
  - WebSpeed Transaction Server 7–20
- Server log filename property
  - AppServer 5–13
  - MS SQL Server DataServer 3–32
  - NameServer 4–6
  - Oracle DataServer 3–19
  - WebSpeed Transaction Server 7–18
- Server logging entry types property
  - AppServer 5–14
  - WebSpeed Transaction Server 7–19
- Server logging level property
  - AppServer agent 5–13
  - AppServer messaging 5–20
  - WebSpeed agent 7–18
  - WebSpeed messaging 7–26
- Server port maximum property 2–31
- Server port minimum property 2–31
- Server startup parameters property
  - AppServer 5–12
  - MS SQL Server DataServer 3–31
  - ODBC DataServer 3–7
  - Oracle DataServer 3–19
  - WebSpeed Transaction Server 7–17
- Server startup timeout property
  - AppServer 5–10
  - WebSpeed Transaction Server 7–15
- Server Transition property 2–31
- Server Transition Timeout property 2–30
- Server watch dog interval property
  - AppServer 5–16
  - WebSpeed Transaction Server 7–21
- ServerGroup group 2–2
- Service name or port property 2–16
- serviceFaultLevel service property A–10
- serviceLoggingLevel service property A–10
- Session connection ID property 7–7
- Session-Free
  - AppServers 5–2
  - Web services 6–4
- Session-Managed
  - AppServers 5–2
  - Web services 6–4
- Shared memory overflow size property 2–11
- Shared memory systems 2–22
- Shutdown property 5–17
- SOAP-AVM gateway 6–2
- SonicMQ Adapters
  - actions allowed on 8–3
  - broker properties 8–4
    - advanced features 8–8
    - AppService Name List 8–6
    - controlling NameServer 8–5
    - general 8–4
    - logging setting 8–7
    - owner information 8–5
  - configuration overview 8–2
  - creating instances 8–14
  - deleting instances 8–15
  - editing a configuration 8–13
  - enabling or disabling for access 8–12
  - environment variables 8–12
  - server properties 8–8
    - general 8–9
    - logging setting 8–9
    - pool range 8–10
  - SSL properties 8–10
    - advanced features 8–11
    - general 8–11
  - starting
    - manually 8–16
  - status 8–18
  - stopping 8–17
  - viewing log files 8–20
- SonicMQ ServerConnect broker log
  - filename property
    - AppServer 5–19
    - WebSpeed Transaction Server 7–25
- SonicMQ ServerConnect enabled property
  - AppServer 5–19
  - WebSpeed Transaction Server 7–25

- SonicMQ ServerConnect server log
    - filename property
      - AppServer 5–20
      - WebSpeed Transaction Server 7–26
  - Source startup arguments property 2–34
  - source.repl.properties file 2–27
  - Spin lock retries property 2–12
  - SSL session cache timeout property
    - AppServer 5–18
    - WebSpeed Transaction Server 7–24
  - staleO4GLObjectTimeout service property A–10
  - Starting
    - AppServers
      - automatically 5–23
      - manually 5–22
    - MS SQL Server DataServers
      - automatically 3–34
      - manually 3–34
    - NameServers
      - automatically 4–16
      - manually 4–16
    - ODBC DataServers
      - automatically 3–10
      - manually 3–9
    - OpenEdge databases
      - automatically 2–21
      - manually 2–20
    - Oracle DataServers
      - automatically 3–21
      - manually 3–21
    - SonicMQ Adapters
      - manually 8–16
    - Web Services Adapters 6–17
    - WebSpeed Transaction Servers
      - automatically 7–29
      - manually 7–28
  - Startup parameters property
    - AppServer 5–17
    - WebSpeed Transaction Server 7–23
  - Startup property 5–17
  - Status
    - AppServer 5–25
    - deployed Web Services 6–26
    - MS SQL Server DataServer 3–35
    - NameServer 4–18
    - ODBC DataServer 3–10
    - OpenEdge database 2–23
    - Oracle DataServer 3–22
    - SonicMQ Adapter 8–18
    - Web Services Adapter 6–22
      - run-time 6–22
  - Stopping
    - AppServers 5–23
    - MS SQL Server DataServers 3–34
    - NameServers 4–17
    - ODBC DataServers 3–10
    - OpenEdge databases 2–20
    - Oracle DataServers 3–22
    - SonicMQ Adapters 8–17
    - Web Services Adapters 6–24
    - WebSpeed Transaction Servers 7–29
  - Structure file containing After image area definitions property 2–35
  - Supports default service property
    - AppServer 5–6
    - MS SQL Server DataServer 3–29
    - ODBC DataServer 3–5
    - Oracle DataServer 3–17
    - WebSpeed Transaction Server 7–11
  - Synchronous or Asynchronous property 2–32
- ## T
- Target startup arguments property 2–34
  - target.repl.properties file 2–27
  - TCP/IP version property
    - AppServer 5–11
    - NameServer 4–8
    - OpenEdge database 2–17
    - WebSpeed Transaction Server 7–16
  - To enable administration roles, select from the following list property 6–13
  - Transition to Agents property 2–34
- ## U
- ubroker.properties file 1–4
  - Unified Broker products 4–2
  - Updating
    - See* Changing
  - URL property 6–9
  - User time statistic 2–24
  - Username property
    - AppServer 5–5
    - MS SQL Server DataServer 3–28
    - ODBC DataServer 3–4
    - Oracle DataServer 3–16
    - WebSpeed Transaction Server 7–10

**V**

## Viewing

## AppServer

- agent properties 5–11
- broker properties 5–4
- environment variables 5–21
- log files 5–28
- messaging properties 5–19
- SSL properties 5–17
- status 5–25

## AppServer Internet Adapter

- log files 5–38

## AppServer Internet Adapter properties 5–31

## deployed Web service

- run-time statistics 6–26
- statistics 6–23
- status 6–26

## MS SQL Server DataServer

- broker properties 3–27
- environment variables 3–32
- log files 3–36
- server properties 3–31
- status 3–35

## NameServer

- environment variables 4–10
- log files 4–21
- properties 4–4
- status 4–18

## ODBC DataServer

- broker properties 3–3
- environment variables 3–8
- log files 3–12
- server properties 3–7
- status 3–10

## OpenEdge database

- basic properties 2–5
- default properties 2–6
- log files 2–3
- process details 2–23
- server group properties 2–16
- status 2–23

## Oracle DataServer

- broker properties 3–15
- environment variables 3–19
- log files 3–24
- server properties 3–18
- status 3–22

## SonicMQ Adapter

- broker properties 8–4
- environment variables 8–12
- log files 8–20
- server properties 8–8
- SSL properties 8–10
- status 8–18

## Web Services Adapter

- advanced features properties 6–14
- location properties 6–8
- logging setting properties 6–11
- properties 6–8

- proxy server setting properties 6–9
- security properties 6–13
- status 6–22

- run-time 6–22

## WSDL properties 6–10

## WebSpeed Messenger

- log files 7–7
- properties 7–2

## WebSpeed Transaction Server

- agent properties 7–16
- broker properties 7–10
- environment variables 7–27
- messaging properties 7–25
- SSL properties 7–23

## Virtual size statistic 2–24

**W**

## waitIfBusy service property A–11

## Watchdog process 2–22

## Watchdog process property 2–9

## Web services

- See* OpenEdge Web Services

## Web Services Adapter

- Admin or Operator role 6–14

## Web Services Adapters

- actions allowed on 6–5
- changing run-time properties 6–23
- configuration overview 6–7
- creating instances 6–5
- deleting instances 6–24
- deploying Web services 6–18
- description 6–2
- editing a configuration 6–16
- enabling or disabling Web services 6–25
- exporting Web services 6–20
- importing Web services 6–20
- listing deployed Web services 6–21
- properties 6–8
  - advanced features 6–14
  - location 6–8
  - logging setting 6–11
  - proxy server setting 6–9
  - security 6–13
  - WSDL 6–10
- setting defaults for Web services 6–17
- starting 6–17
- status 6–22
  - run-time 6–22
- stopping 6–24
- undeploying Web services 6–30
- updating Web services 6–27
- viewing Web service statistics 6–23
- Web service run-time statistics 6–26

## Web Services Protocol Value property 6–9

- WebSpeed
  - additional management 7–32
- WebSpeed Messengers
  - configuration overview 7–2
  - editing a configuration 7–7
  - properties 7–2
    - advanced features 7–6
    - controlling NameServer 7–3
    - general 7–3
    - logging setting 7–5
    - SSL 7–6
    - WebSpeed Transaction Server
      - Location 7–4
  - viewing log files 7–7
- WebSpeed transaction server host property 7–4
- WebSpeed transaction server port property 7–4
- WebSpeed Transaction Servers
  - actions allowed on 7–9
  - additional management 7–32
  - agent properties 7–16
    - advanced features 7–22
    - general 7–17
    - logging setting 7–18
    - pool range 7–21
  - agents
    - adding or trimming 7–29
  - broker properties 7–10
    - advanced features 7–15
    - AppService Name List 7–11
    - controlling NameServer 7–11
    - general 7–10
    - logging setting 7–12
    - owner information 7–10
    - configuration overview 7–9
    - creating instances 7–28
    - deleting instances 7–31
    - editing a configuration 7–27
    - environment variables 7–27
    - messaging properties 7–25
    - SSL properties 7–23
      - advanced features 7–24
      - general 7–24
    - starting
      - automatically 7–29
      - manually 7–28
    - stopping 7–29
- Weighted CPU statistic 2–24
- Working directory property
  - AppServer 5–5
  - MS SQL Server DataServer 3–27
  - NameServer 4–5
  - ODBC DataServer 3–3
  - Oracle DataServer 3–15
  - Web Services Adapter 6–11
  - WebSpeed Messenger 7–3
  - WebSpeed Transaction Server 7–10
- WSA
  - See* Web Services Adapter
- WSA administration authentication property 6–13
- WSA administration SOAP action property 6–14
- WSDL inaccessible page property 6–10
- WSDL listing page property 6–10

