

OpenEdge® Getting Started: Installation and Configuration

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation. The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document. The references in this manual to specific platforms supported are subject to change.

A (and design), Actional, Actional (and design), Affinities Server, Allegrix, Allegrix (and design), Apama, Business Empowerment, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Connect OLE DB, DataDirect Technologies, DirectAlert, Dynamic Index Utility, Dynamic Routing Architecture, EasyAsk, EdgeXtend, Empowerment Center, eXcelon, Fathom, Halo, IntelliStream, Iwawe Integrator, LiveMine, Madrid, Neon, Neon 24x7, Neon New Era of Networks, Neon Unload, O (and design), ObjectStore, OpenEdge, PDF, PeerDirect, Persistence, Persistence (and design), POSSENET, Powered by Progress, PowerTier, ProCare, Progress, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress Fast Track, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, ProtoSpeed, ProVision, SequeLink, Shadow, ShadowDirect, Shadow Interface, Shadow Web Interface, Shadow Web Server, Shadow TLS, Sonic ESB, SonicMQ, SOAPStation, Sonic Software (and design), SonicSynergy, Speed Load, SpeedScript, Speed Unload, Stylus Studio, Technical Empowerment, UIM/X, Visual Edge, Voice of Experience, WebSpeed, and Your Software, Our Technology—Experience the Connection are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, Connect Everything, Achieve Anything., DataDirect Spy, DataDirect SupportLink, DataDirect XQuery, DataXtend, Future Proof, Ghost Agents, GVAC, Looking Glass, ObjectCache, ObjectStore Inspector, ObjectStore Performance Expert, Pantero, POSSE, ProDataSet, Progress DataXtend, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, PSE Pro, PS Select, 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 Orchestration Server, Sonic XML Server, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries. Vermont Views is a registered trademark of Vermont Creative Software in the U.S. and other countries. IBM is a registered trademark of IBM Corporation. JMX and JMX-based marks and Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners.

Third Party Product Notices for OpenEdge

OpenEdge includes Imaging Technology copyrighted by Snowbound Software 1993-2003. www.snowbound.com.

OpenEdge includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xerces C++ Parser (XML) and Xerces2 Java Parser (XML)); Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces Parser (XML)); and Copyright © 2000-2003 The Apache Software Foundation. All rights reserved (Ant). The names “Apache,” “Xerces,” “ANT,” 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”, nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact apache@apache.org. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes software developed by Vermont Creative Software. Copyright © 1988-1991 by Vermont Creative Software.

OpenEdge includes software developed by IBM and others. Copyright © 1999, International Business Machines Corporation and others. All rights reserved.

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

OpenEdge includes the UnixWare platform of Perl Runtime authored by Kiem-Phong Vo and David Korn. Copyright © 1991, 1996 by AT&T Labs. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED “AS IS”, WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T LABS MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

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

OpenEdge includes software developed by the World Wide Web Consortium. Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All rights reserved. This work is distributed under the W3C® Software License [<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>] in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

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

OpenEdge includes the JMX Technology from Sun Microsystems, Inc.

OpenEdge includes software developed by the ModelObjects Group (<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 this software without prior written permission. Products derived from this 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.

OpenEdge 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.

December 2006



Contents

Preface	Preface–1
----------------------	------------------

Part I Installation

1. Windows Platforms Installation Requirements	1–1
System requirements	1–2
Ensuring you have the most up-to-date system requirements information	1–2
Java considerations	1–2
Windows 2000, Windows 2003, and Windows XP Professional	1–3
Server compatibility	1–5
OpenEdge RDBMS compatibility	1–5
OpenEdge DataServer compatibility	1–6
OpenEdge SQL Database Server	1–7
AppServer compatibility	1–7
Transaction server compatibility	1–9
Required third-party applications	1–10
Products that require Microsoft .NET Framework	1–10
Installing the DataDirect SQL ODBC drivers and the DataDirect ODBC branded drivers	1–11
Platform Support for OpenEdge Management 3.1B	1–14

2. UNIX Systems Installation Requirements	2-1
UNIX system requirements	2-2
Ensuring you have the most up-to-date system requirements information	2-2
Java requirements	2-2
Ensuring you have the most up-to-date Java requirements information.	2-6
Requirements for building ABL applications.	2-6
Requirements for running OpenEdge applications.	2-7
ABL client/database server compatibility	2-8
Supported platforms	2-8
Support for Linux PowerPC	2-9
Support for Failover Clusters	2-10
Expanded platform support for OpenEdge Management.	2-10
JDBC Type 4 driver support on 64-bit platforms	2-11
Licensing	2-12
 3. OpenEdge Installation Prerequisites.	 3-1
Tasks overview	3-2
Gathering information to plan your installation	3-3
Determining your installation method	3-5
Determining the type of installation	3-6
How product selection can affect your installation tasks	3-7
Shared Network Installation Utility	3-7
Disk space requirements for OpenEdge installed products	3-8
OpenEdge products installed in Windows	3-8
OpenEdge products installed on UNIX.	3-10
Windows-specific installation considerations	3-12
OpenEdge working directory reminder.	3-13
Ensuring .dll and .ocx files in the Windows directory are read-only.	3-13
Software required to run OpenEdge products or components.	3-13
Saving an existing OpenEdge or Progress installation in Windows-based platforms	3-14
Before you begin	3-15
Reviewing the Windows platform installation directory structure	3-17
Integrating OpenEdge with Windows Explorer.	3-21
UNIX-specific installation considerations	3-23
Specific considerations.	3-23
Upgrading an existing OpenEdge or Progress installation on UNIX platforms.	3-24
Reviewing the UNIX system installation directory structure.	3-25

Installing or viewing product documentation and samples	3-28
Installing documentation and samples in Windows platforms.....	3-28
Installing documentation and samples on UNIX platforms	3-30
WebSpeed configuration choices	3-32
4. Performing an OpenEdge Installation in Windows.....	4-1
Installation overview	4-2
Loading the installation media	4-2
Performing the installation.....	4-3
Finishing the installation	4-4
Postinstallation considerations	4-5
Running the Progress Dynamics Configuration Utility	4-5
Before you begin.....	4-6
Completing the DCU wizard	4-7
Editing Progress Dynamics files	4-13
Editing installed files	4-15
Editing the Progress Dynamics XML configuration file	4-16
Starting a development session	4-19
Stopping and restarting Progress Dynamics.....	4-20
Updating session types.....	4-21
Running the Entity Import tool.....	4-23
Recompiling application code	4-24
Setting up for Web development.....	4-25
Additional product installation activities	4-25
Installing additional products.....	4-26
Installing additional components to previously installed products.....	4-27
Viewing registry information	4-29
Downloading executables for heterogeneous environments	4-30
OpenEdge Silent installation overview	4-31
Data input options for a Silent installation.....	4-32
Understanding the Response.ini file contents.....	4-33
Running the Silent installation.....	4-37
Checking the status of the Silent installation log file	4-38
Optional data input activities	4-40
Performing postinstallation tasks	4-41
Uninstalling OpenEdge in Windows	4-43
Using the Uninstall or Add/Remove Programs utility	4-43
Manually removing OpenEdge	4-46
Sharing an OpenEdge installation on a network overview	4-50
Primary tasks	4-50
Networking overview.....	4-51
Determining a shared network to clients connection.....	4-51
Setting up the shared network	4-52

Running the Shared Network Installation Utility to set up a client connection	4-53
Reviewing local intranet security settings.	4-56
Uninstalling the Shared Network Installation Utility	4-57
Running the Silent installation option for the Shared Network Installation Utility . .	4-58
5. Performing an OpenEdge Installation on UNIX	5-1
Installation overview	5-2
Loading the installation media	5-2
Performing the installation	5-6
Finishing the installation	5-7
Additional product installation activities	5-7
Installing additional products	5-8
Adding components to previously installed products	5-9
Downloading executables for heterogeneous environments	5-13
OpenEdge Silent installation overview	5-13
Data input options for a Silent installation	5-14
Understanding the Response.ini file contents	5-15
Running the Silent installation	5-18
Checking the status of the Silent Installation log file	5-20
Optional data input activities	5-21
Performing postinstallation tasks	5-22
Required tasks	5-22
Setting AdminServer security	5-23
Uninstalling OpenEdge on UNIX and Linux operating systems	5-24
Uninstalling OpenEdge Adapter for SonicMQ and OpenEdge Adapter for SonicESB	5-24
6. Administration Utilities	6-1
Using the License Update utility	6-2
Changes to accommodate license updates	6-2
Displaying license information using the SHOWCFG utility	6-5
Using the SHOWCFG utility in Windows platforms	6-5
Using the SHOWCFG utility on UNIX or Linux platforms	6-7
Displaying license information in Windows platforms	6-8
Managing user licenses on all supported platforms	6-9
OpenEdge license information	6-9
Using the OpenEdge license file	6-10
Using OpenEdge resources in Windows platforms	6-12
Shared memory	6-12
Processes in Windows platforms	6-13
Manage memory and system configurations on UNIX platforms	6-14
Calculating memory needs	6-14
Managing shared memory and process resources	6-18

Reducing memory usage	6-19
Swap space	6-19
Shared memory and kernel configuration	6-20
UNIX troubleshooting tips	6-22
Error messages	6-23
Altered or missing progress.cfg file	6-23
Tailoring startup scripts	6-24
OpenEdge event logging	6-25
OpenEdge event log file	6-25
Managing the OpenEdge event log file size	6-26
Event logging in Windows	6-28

Part II Configuration

7. Working in the OpenEdge Environment in Windows	7-1
Reviewing environment variables	7-3
System environment variables	7-3
Java environment variables	7-4
Windows registry and the progress.ini file	7-4
Additional details for Java-related environment variables	7-10
Setting OpenEdge Program Item properties	7-11
Using the Proenv utility	7-11
Getting started with the AdminServer	7-11
OpenEdge products supported by the AdminServer	7-12
AdminServer considerations	7-13
AdminServer group name conventions and restrictions	7-13
Creating an OpenEdge database	7-14
Configuring an OpenEdge Database server	7-14
Running OpenEdge	7-15
Maintaining OpenEdge and Progress	7-15
OpenEdge key and certificate stores	7-16
8. Working in the OpenEdge Environment on UNIX	8-1
Default environment variables settings	8-2
UNIX environment variables	8-2
Setting Java environment variables	8-9
Setting the JDK environment variable	8-9
Setting SQL client environment variables	8-10
Using the Proenv utility	8-10
Getting started with the AdminServer	8-10
AdminServer considerations	8-11
How to implement the User-Group Authorization feature	8-12

Understanding the built-in terminal definitions	8-13
Terminal issues	8-13
Terminal identifiers	8-14
Additional terminal identifier considerations	8-15
OpenEdge key and certificate stores	8-16
9. Managing OpenEdge Key and Certificate Stores	9-1
Managing key stores for OpenEdge servers	9-2
Establishing a trusted SSL server identity	9-3
Using pkiutil to manage an OpenEdge key store	9-5
Understanding key store content	9-6
Using genpassword to obtain a key store password-encrypted value	9-8
Managing certificate stores for OpenEdge clients	9-9
Installing trusted CA/root certificates	9-9
Using certutil to manage an OpenEdge root certificate store	9-11
Using mkhashfile to install root certificates in the OpenEdge root certificate store	9-12
10. Configuration	10-1
Overview of the Progress Explorer Framework	10-2
Introduction	10-2
Framework elements and descriptions	10-3
Additional Progress Explorer Framework Considerations	10-9
Working with Unified Brokers	10-9
Running locally	10-9
Running remotely	10-10
Unified Broker common elements	10-10
Using default sample brokers	10-11
Configuring and starting Unified Broker instances	10-13
Understanding and Using the AdminServer	10-16
Starting the AdminServer	10-16
Stopping the AdminServer	10-18
Changing the default port	10-19
Changing the startup setting (in Windows only)	10-20
Running more than one AdminServer	10-20
Querying the AdminServer	10-21
AdminServer-related authorization option	10-22
Using the Progress Explorer tool	10-23
OpenEdge Servers that Progress Explorer supports	10-23
Saving configurations	10-25

Mergeprop utility overview	10–26
Property value	10–26
Using the mergeprop utility	10–27
Logical structure and syntax of property files	10–35
Ubroker.properties file and product configurations	10–38
Unified Broker products and associated clients	10–39
Unified Broker installation prerequisites	10–39
Ubroker.properties file structure	10–40
Command-line utilities reference	10–43
 11. Starting and Running OpenEdge	11–1
Starting OpenEdge in Windows platforms	11–2
Startup and shutdown	11–2
Starting OpenEdge as a Windows service	11–5
Starting single-user OpenEdge in interactive mode	11–6
Starting single-user OpenEdge in batch or background mode	11–7
Starting the multi-user server or broker.	11–7
Starting the multi-user server or broker as a Windows service	11–8
Starting OpenEdge on UNIX platforms	11–9
Startup and shutdown commands.	11–9
Starting single-user OpenEdge in interactive mode	11–11
Starting single-user OpenEdge in batch or background mode	11–12
Starting the multi-user server or broker.	11–13
Running OpenEdge clients and servers on a network	11–13
Using network startup parameters	11–13
Specifying the network type (-N)	11–15
Network addressing (-S and -H)	11–16
Starting applications on a network	11–16
Starting multiple brokers using the same protocol	11–18
Accessing a server behind a firewall.	11–19
Starting and running multi-user OpenEdge in interactive mode in Windows	11–20
Starting and running multi-user OpenEdge in interactive mode on UNIX	11–21
Starting and running multi-user OpenEdge clients in batch or background mode in Windows	11–22

Part III OpenEdge Products and Components

12. OpenEdge Installation Products and Components in Windows	12–1
OpenEdge Release product headlights	12–2
OpenEdge Architect	12–2

Audit Policy Maintenance (APM)	12-2
OpenEdge Adapter for Sonic ESB	12-2
Web Services Adapter	12-3
OpenEdge installation options	12-4
Complete installation option	12-4
Custom installation option	12-4
OpenEdge product components and subcomponents	12-5
4GL Development System	12-5
AppServer Internet Adapter (AIA)	12-9
Client Networking	12-10
NameServer	12-14
NameServer Load Balancer	12-15
OpenEdge Adapter for Sonic ESB	12-15
OpenEdge Application Server — Basic	12-16
OpenEdge Application Server — Enterprise	12-19
OpenEdge Architect	12-23
OpenEdge DataServer for MS SQL Server	12-29
OpenEdge DataServer for ODBC	12-31
OpenEdge DataServer for Oracle	12-34
OpenEdge Development Server	12-36
OpenEdge Enterprise RDBMS	12-41
OpenEdge Personal RDBMS	12-45
OpenEdge Replication	12-49
OpenEdge Replication Plus	12-50
OpenEdge SQL Client Access	12-51
OpenEdge Studio	12-52
OpenEdge Workgroup RDBMS	12-57
Query/Results	12-61
Translation Manager	12-63
Visual Translator	12-64
Web Services Adapter	12-66
WebSpeed Messenger	12-66
WebSpeed Workshop	12-67
13. OpenEdge Installation Products and Components on UNIX	13-1
OpenEdge Product highlights	13-2
Audit Policy Maintenance	13-2
OpenEdge Adapter for Sonic ESB	13-2
Web Services Adapter	13-3
OpenEdge installation options	13-4
Complete installation option	13-4
Custom installation option	13-4
OpenEdge product components and subcomponents	13-5
4GL Development System	13-5

AppServer Internet Adapter (AIA)	13-9
Client Networking	13-10
OpenEdge Replication	13-13
OpenEdge Replication Plus	13-13
NameServer	13-14
NameServer Load Balancer	13-15
OpenEdge Adapter for SonicESB	13-15
OpenEdge Application Server — Basic	13-16
OpenEdge Application Server — Enterprise	13-18
OpenEdge DataServer for Oracle	13-21
OpenEdge Development Server	13-24
OpenEdge Enterprise RDBMS	13-28
OpenEdge Personal RDBMS	13-32
OpenEdge Workgroup RDBMS	13-35
OpenEdge SQL Client Access	13-38
Query/Results	13-39
WebSpeed Messenger	13-41
Web Services Adapter	13-42
 A. Preinstallation Checklist for Windows	 A-1
Before you start	A-2
Products to install	A-2
Prerequisite third-party software	A-2
Values from your existing OpenEdge installation	A-3
Installation and working directories	A-3
Installation type	A-3
Recommended and optional components	A-4
Database	A-5
OpenEdge Adapter for Sonic ESB	A-5
Web server	A-6
Progress Dynamics	A-7
Language in which online messages appear	A-8
Character set, date, and number formats	A-8
Web Services Adapter (WSA)	A-9
Options to secure your AdminServer	A-10
 B. Preinstallation Checklist for UNIX	 B-1
Before you start	B-2
Java platform requirements	B-2
Products to install	B-2
Values from your existing OpenEdge installation	B-2
Installation and working directories	B-3
Installation type	B-3
Database	B-3

OpenEdge Adapter for Sonic ESB	B-3
Web server	B-4
Language in which messages appear	B-5
Character set, date, and number formats.	B-6
Web Services Adapter (WSA)	B-6
OpenEdge product scripts and program modules	B-7
Identical file exists in the installation directory	B-7
C. Command and Utility Reference	C-1
Administering and configuring Unified Broker products	C-2
ASBMAN	C-2
DBMAN	C-4
Mergeprop	C-5
NSMAN	C-8
PROADSV	C-10
WTBMAN	C-13
Installing and managing keys and digital certificates	C-15
certutil	C-15
genpassword	C-17
mkhashfile	C-17
pkiutil	C-18
D. OpenEdge National Language Support	D-1
Packaging	D-2
Directory structure	D-3
Contents of each directory	D-4
Implementing regional support	D-5
International databases	D-5
Progress messages	D-6
File protection.	D-6
Environment variables of the SQL client	D-11
Regional parameter files	D-12
Progress.ini file and the Windows registry	D-14
E. NameServer and Name Server Load Balancing Details	E-1
NameServer overview	E-2
Unified Broker and Name Server relationship	E-3
Configuring NameServer communications	E-4
Understanding load balancing	E-5
Understanding server-level and connection-level fault tolerance	E-7
Connection-level fault tolerance.	E-8
Using UDP broadcasting	E-9
Using NameServer replication	E-10

Using NameServer neighbors	E-13
Performance implications of broadcasting	E-15
Configuring OpenEdge NameServer instances	E-16
Downloading NameServer executables	E-16
Order of configuration	E-16
Configuring and using NameServer instances	E-17
Configuring the NameServer in the Progress Explorer	E-18
Starting and managing a NameServer using the Progress Explorer ...	E-19
F. Configuration Models	F-1
Shared-memory configurations	F-2
Shared-memory architecture	F-3
Client/server configurations	F-5
Terminology	F-5
Simple client/server configurations	F-7
Client/server and OpenEdge AppServer in the network environment	F-9
Preparing to run OpenEdge in a TCP/IP network	F-15
Installing OpenEdge on your TCP/IP network	F-15
Typical TCP/IP configuration with a hard disk on each machine	F-16
Setting up network files to run OpenEdge	F-17
Configuring OpenEdge on a network operating system	F-17
G. AdminServer Authorization and Authentication	G-1
AdminServer logging details	G-2
Log format	G-2
Determine the data logged in the AdminServer log	G-4
Setting authentication option to start servers administered by the AdminServer	G-5
Index	Index-1

Tables

Table 1–1:	Windows system requirements to run OpenEdge	1–3
Table 1–2:	Client/Database server compatibility	1–5
Table 1–3:	Client/OpenEdge SQL Database Server	1–7
Table 1–4:	Client/AppServer compatibility	1–8
Table 1–5:	.Net Client/AppServer compatibility	1–8
Table 1–6:	Transaction Server compatibility	1–9
Table 1–7:	Driver files — Windows operating systems for SQL	1–11
Table 1–8:	Driver files — Windows operating systems with OpenEdge DataServer for ODBC	1–12
Table 1–9:	Data-source components and version numbers	1–13
Table 2–1:	JRE/JDK requirements by platform	2–5
Table 2–2:	Minimum requirements for running OpenEdge applications	2–7
Table 2–3:	Client/server compatibility	2–8
Table 2–4:	Supported platforms	2–8
Table 3–1:	Preinstallation documentation resources	3–3
Table 3–2:	Installation options	3–6
Table 3–3:	Disk space requirements for OpenEdge installed products in Windows	3–8
Table 3–4:	Disk space requirements for OpenEdge products categorized by platforms	3–11
Table 3–5:	OpenEdge-install-dir (%DLC%) directory structure	3–18
Table 3–6:	OpenEdge-install-dir (\$DLC) directory structure	3–25
Table 4–1:	Progress Dynamics files that you can edit	4–15
Table 4–2:	Managers for customized session types	4–21
Table 4–3:	Data input options for a Silent installation	4–32
Table 4–4:	Available Network Server and Client Shortcuts	4–55
Table 5–1:	Data input options for a Silent installation	5–14
Table 5–2:	User-group parameter options	5–23
Table 6–1:	Running the SHOWCFG utility	6–5
Table 6–2:	Display fields associated with the SHOWCFG utility	6–6
Table 6–3:	Components used to calculate memory needs	6–15
Table 6–4:	Size increments for increasing startup parameters by 1	6–16
Table 6–5:	Single-user memory requirements	6–17
Table 6–6:	Multi-user memory requirements	6–17
Table 6–7:	Formulas for calculating memory requirements	6–17
Table 6–8:	Shared memory and semaphore parameter settings	6–20
Table 6–9:	Error codes and kernel reconfiguration parameters	6–21
Table 6–10:	Error messages	6–23
Table 6–11:	Reasons for altered or missing progress.cfg file	6–23
Table 6–12:	Progress event logging components	6–28
Table 6–13:	Event level values	6–29
Table 6–14:	Windows application event log components	6–31
Table 7–1:	Supported environment variables	7–6

Table 8–1:	UNIX environment variables	8–4
Table 8–2:	User-Group parameter options	8–12
Table 8–3:	Terminal identifiers	8–14
Table 10–1:	Elements of the Progress Explorer Framework	10–5
Table 10–2:	Default sample broker for each Unified Broker product	10–11
Table 10–3:	Command line input to the mergeprop command	10–28
Table 10–4:	Property files managed by the mergeprop utility	10–30
Table 10–5:	New value formats supported in all property files	10–36
Table 10–6:	New value formats supported in mergeprop delta files only	10–37
Table 10–7:	Value formats supported prior to OpenEdge 10	10–37
Table 10–8:	Unified Broker products and the clients that they support	10–39
Table 10–9:	Command-line utilities to start and stop installed OpenEdge products	10–43
Table 10–10:	Command-line utilities to validate property files	10–44
Table 11–1:	Windows GUI startup and shutdown commands	11–3
Table 11–2:	Windows startup and shutdown commands	11–4
Table 11–3:	OpenEdge command components	11–9
Table 11–4:	UNIX startup and shutdown commands	11–10
Table 11–5:	Client network parameters	11–14
Table 11–6:	Server network parameters	11–14
Table 11–7:	Default network types	11–15
Table 12–1:	4GL Development System components and subcomponents	12–5
Table 12–2:	AppServer Internet Adapter (AIA) components and subcomponents	12–9
Table 12–3:	Client Networking components and subcomponents	12–10
Table 12–4:	NameServer components and subcomponents	12–14
Table 12–5:	NameServer Load Balancer components and subcomponents	12–15
Table 12–6:	OpenEdge Adapter for Sonic ESB components and subcomponents	12–15
Table 12–7:	OpenEdge Application Server — Basic components and subcomponents	12–16
Table 12–8:	OpenEdge Application Server — Enterprise components and subcomponents	12–19
Table 12–9:	OpenEdge Architect components and subcomponents	12–23
Table 12–10:	OpenEdge DataServer — MS SQL Server components and subcomponents	12–29
Table 12–11:	OpenEdge DataServer — ODBC components and subcomponents ..	12–31
Table 12–12:	OpenEdge DataServer for Oracle components and subcomponents ..	12–34
Table 12–13:	OpenEdge Development Server components and subcomponents ..	12–36
Table 12–14:	OpenEdge Enterprise RDBMS components and subcomponents ...	12–41
Table 12–15:	OpenEdge Personal RDBMS components and subcomponents	12–45
Table 12–16:	OpenEdge Replication components and subcomponents	12–49
Table 12–17:	OpenEdge Replication Plus components and subcomponents	12–50
Table 12–18:	OpenEdge SQL Client Access components and subcomponents	12–51
Table 12–19:	OpenEdge Studio components and subcomponents	12–52

Table 12–20:	OpenEdge Workgroup components and subcomponents	12–57
Table 12–21:	Query/Results components and subcomponents	12–61
Table 12–22:	Translation Manager components and subcomponents	12–63
Table 12–23:	Visual Translator components and subcomponents	12–64
Table 12–24:	Web Services Adapter components and subcomponents	12–66
Table 12–25:	WebSpeed Messenger components and subcomponents	12–66
Table 12–26:	WebSpeed Workshop components and subcomponents	12–67
Table 13–1:	4GL Development System components and subcomponents	13–5
Table 13–2:	AppServer Internet Adapter components and subcomponents	13–9
Table 13–3:	Client Networking components and subcomponents	13–10
Table 13–4:	OpenEdge Replication components and subcomponents	13–13
Table 13–5:	OpenEdge Replication Plus components and subcomponents	13–13
Table 13–6:	NameServer components and subcomponents	13–14
Table 13–7:	NameServer Load Balancer components and subcomponents	13–15
Table 13–8:	SonicMQ Adapter components and subcomponents	13–15
Table 13–9:	OpenEdge Application Server — Basic components and subcomponents	13–16
Table 13–10:	Application Server — Enterprise components and subcomponents ...	13–18
Table 13–11:	OpenEdge DataServer for Oracle components and subcomponents ..	13–21
Table 13–12:	OpenEdge Development Server components and subcomponents ...	13–24
Table 13–13:	OpenEdge Enterprise RDBMS components and subcomponents ...	13–28
Table 13–14:	OpenEdge Personal RDBMS components and subcomponents	13–32
Table 13–15:	OpenEdge Workgroup RDBMS components and subcomponents ...	13–35
Table 13–16:	OpenEdge SQL Client Access components and subcomponents ...	13–38
Table 13–17:	Query/Results components and subcomponents	13–39
Table 13–18:	WebSpeed Messenger components	13–41
Table 13–19:	Web Services Adapter components and subcomponents	13–42
Table C–1:	Command line input to the mergeprop command	C–7
Table C–2:	proadsv command-line options	C–12
Table D–1:	PROMSGS translations shipped with OpenEdge	D–2
Table D–2:	Supplemental PROMSGS translations available for download	D–3
Table D–3:	National language file descriptions	D–4
Table D–4:	PROMSGS file synchronization process	D–8
Table D–5:	Example of PROMSGS files being out of sync	D–10
Table D–6:	Startup parameters for a deployed application	D–12
Table D–7:	Environment variables	D–15
Table E–1:	Weight factors based on percentage	E–6
Table E–2:	Weight factors based on arbitrary sums	E–6
Table F–1:	TCP/IP network files	F–17

Figures

Figure 6–1:	OpenEdge Configuration Information from SHOWCFG utility	6–6
Figure 6–2:	Product configuration details display using SHOWCFG utility	6–8
Figure 6–3:	Windows application event log components	6–30
Figure 6–4:	Windows application Event Properties dialog box	6–32
Figure 10–1:	Overview of the Progress Explorer Framework	10–4
Figure E–1:	Sample Unified Broker client services file	E–5
Figure E–2:	Server-level and connection-level fault tolerance	E–7
Figure E–3:	NameServer replication	E–10
Figure E–4:	NameServer neighbors	E–14
Figure F–1:	Shared-memory OpenEdge architecture	F–3
Figure F–2:	Simple client/server configuration	F–7
Figure F–3:	Multiple system client/server configuration	F–8
Figure F–4:	Simple OpenEdge network configuration	F–10
Figure F–5:	Network file server for application files	F–11
Figure F–6:	Network file server as a database server	F–12
Figure F–7:	Network file server for application and database files	F–13
Figure F–8:	LAN configuration with the OpenEdge AppServer	F–14
Figure F–9:	Typical TCP/IP configuration (file server not used)	F–16

Preface

This Preface contains the following sections:

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

Purpose

This book describes installation and configuration topics related to the Release 10.1B for the following operating systems:

- Windows 2000, Windows 2003, and Windows XP Professional for the Intel architecture.

Unless otherwise noted, platform references throughout this guide have been simplified for readability. Windows refers to Windows 2000, Windows 2003, and Windows XP Professional.

Note: Virtualization software, such as Citrix Presentation Server and VMware, are not described in this manual. Details about virtualization software support are documented in *OpenEdge 10 Platform and Product Availability Guide*.

- UNIX and Linux.

Unless otherwise noted, platform references throughout this guide have been simplified for readability. UNIX refers to UNIX and Linux.

Audience

Administrative and technical personnel responsible for installing and configuring OpenEdge® Release 10.1B.

Organization

Part I, Installation

Chapter 1, “Windows Platforms Installation Requirements”

Lists system and platform prerequisites and requirements for installing OpenEdge in Windows.

Chapter 2, “UNIX Systems Installation Requirements”

Lists system and platform prerequisites and requirements for installing OpenEdge on UNIX/Linux.

Chapter 3, “OpenEdge Installation Prerequisites”

Identifies prerequisite information to know and preliminary tasks to perform before you install OpenEdge software in Windows or on UNIX.

Chapter 4, “Performing an OpenEdge Installation in Windows”

Contains information related to installation and postinstallation tasks for OpenEdge in Windows platforms. (The detailed procedures to complete the Installation Utility are presented only in the Windows online help.)

Chapter 5, “Performing an OpenEdge Installation on UNIX”

Contains information related to installation and postinstallation tasks when installing OpenEdge on UNIX platforms. (The detailed procedures to complete the Installation Utility are presented only in the UNIX online help.)

Chapter 6, “Administration Utilities”

Provides step-by-step instructions to perform a variety of administrative tasks and describes how to manage Windows and UNIX platform-specific resources, respectively.

Part II, Configuration

Chapter 7, “Working in the OpenEdge Environment in Windows”

Explains how the OpenEdge environment works in Windows.

Chapter 8, “Working in the OpenEdge Environment on UNIX”

Explains how the OpenEdge environment works on UNIX.

Chapter 9, “Managing OpenEdge Key and Certificate Stores”

Describes how to use OpenEdge utilities to manage key stores for OpenEdge servers and manage certificate stores for OpenEdge clients.

Chapter 10, “Configuration”

Introduces the Progress Explorer Framework, a common administrative architecture for installed OpenEdge server products, and highlights the framework’s elements, focusing on Unified Brokers.

Chapter 11, “Starting and Running OpenEdge”

Provides instructions to start and connect to an OpenEdge RDBMS in different modes. Also provides information about running OpenEdge clients and servers on a network.

Part III, OpenEdge Products and Components

Chapter 12, “OpenEdge Installation Products and Components in Windows”

Identifies the components and subcomponents associated with each product that can be installed in a Windows environment.

Chapter 13, “OpenEdge Installation Products and Components on UNIX”

Identifies the components and subcomponents associated with each product that can be installed in a UNIX environment.

Appendix A, “Preinstallation Checklist for Windows”

Provides a planning tool to determine and record product installation choices in Windows **before** running the OpenEdge Release 10.1B Installation Utility.

Appendix B, “Preinstallation Checklist for UNIX”

Provides a planning tool to determine and record product installation choices on UNIX **before** running the OpenEdge Release 10.1B Installation Utility.

Appendix C, “Command and Utility Reference”

Describes commands and utilities whose primary syntax and functional documentation is in this manual.

Appendix D, “OpenEdge National Language Support”

Provides information about OpenEdge messages.

Appendix E, “NameServer and Name Server Load Balancing Details”

Presents additional detailed information about the NameServer load balancing feature.

Appendix F, “Configuration Models”

Provides information about different configuration models you can reference and details about running OpenEdge installations in a network environment.

Appendix G, “AdminServer Authorization and Authentication”

Provides additional information to use the Adminserver to determine the data logged in the AdminServer log and to set authentication to start servers administered by the AdminServer.

Using this manual

The main topics presented in this guide also work with or point to related installation or configuration details presented in other OpenEdge documentation.

Installation planning and performing

Familiarize yourself with the installation information and tasks for your operating system by proceeding as follows:

- Read the chapters in [Part I, Installation](#) in chronological order to help you plan and perform your installation.
- Reference the information provided in [Part III, OpenEdge Products and Components](#) that provides preinstallation checklists, and product component and subcomponent details.
- Obtain a copy of the installation online help. Reference [Table 3–1](#) for detailed information about where you can locate a copy of the online help.

Configuration concepts

[Part II, Configuration](#) presents general OpenEdge configuration concepts and arrangements. Reference details presented in these chapters:

- [Chapter 9, “Managing OpenEdge Key and Certificate Stores”](#)
- [Chapter 10, “Configuration”](#)

As needed, these chapters point to other product documentation for configuration details.

References to ABL compiler and run-time features



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.

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."

Typographical conventions

This manual uses the following typographical conventions:

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

Convention	Description
[]	Large brackets indicate the items within them are optional.
[]	Small brackets are part of ABL.
{ }	Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.
{ }	Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.
	A vertical bar indicates a choice.
...	Ellipses indicate repetition: you can choose one or more of the preceding items.

Examples of syntax descriptions

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

Syntax

```
ACCUM aggregate expression
```

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

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

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

Syntax

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

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

Syntax

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

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

Syntax

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

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

Syntax

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

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

Syntax

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

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

Syntax

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

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

Syntax

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

Long syntax descriptions split across lines

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

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

Syntax

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

Complex syntax descriptions with both required and optional elements

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

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

Syntax

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

OpenEdge messages

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

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

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

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

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

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

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

Obtaining more information about OpenEdge messages

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

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

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



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

1. Start the Procedure Editor:

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

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

Part I

Installation

[Chapter 1, Windows Platforms Installation Requirements](#)

[Chapter 2, UNIX Systems Installation Requirements](#)

[Chapter 3, OpenEdge Installation Prerequisites](#)

[Chapter 4, Performing an OpenEdge Installation in Windows](#)

[Chapter 5, Performing an OpenEdge Installation on UNIX](#)

[Chapter 6, Administration Utilities](#)

Windows Platforms Installation Requirements

This chapter describes information you need to know before installing OpenEdge® software on any of these supported Windows-based platforms: Windows 2000, Windows 2003, and Windows XP Professional for the Intel 86 architecture, as detailed in the following sections:

- [System requirements](#)
- [Server compatibility](#)
- [Required third-party applications](#)
- [Platform Support for OpenEdge Management 3.1B](#)

System requirements

This section describes the hardware and software requirements for running OpenEdge® Release 10.1B on the following supported platforms:

- Windows 2000
- Windows 2003
- Windows XP Professional for the Intel 86 architecture

Note: The specific support requirements for virtualization software, such as Citrix Presentation Server and VMware, is determined by and dependent upon the vendor's support for the underlying operating system. For more information about virtualization software, see *OpenEdge Platform and Product Availability Guide*.

Ensuring you have the most up-to-date system requirements information

The system requirements provided in this chapter are current as of the publication date of this manual; however, requirements can change. To ensure that you have the most up-to-date information about system requirements, refer to *OpenEdge Release Notes* available with your installation medium and the *OpenEdge 10 Platform & Product Availability Guide* on the Progress Software Corporation Web site

<http://www.progress.com/products/lifecycle/index.ssp>.

Java considerations

Many OpenEdge products require the Java Run-time Environment (JRE), the Java Development Kit (JDK), or both of these components to use specific product functionality once the products are installed. However, the OpenEdge installation automatically installs any needed JDK/JRE components on any of the supported Windows platforms. For specific information about these components, see the *OpenEdge 10 Platform & Product Availability Guide* on the Progress Software Corporation Web site

<http://www.progress.com/products/lifecycle/index.ssp>.

Windows 2000, Windows 2003, and Windows XP Professional

Table 1–1 lists the minimum requirements for running OpenEdge in Window 2000, Windows 2003, and Windows XP Professional for the Intel 86 architecture.

Table 1–1: Windows system requirements to run OpenEdge *(1 of 2)*

Component	Requirement
ABL client	<p>A Pentium PC (or compatible computer) with these characteristics:</p> <ul style="list-style-type: none"> • Clock speed — A minimum of 100MHZ. • Memory — A minimum of 32MB.
OpenEdge database server	<p>A Pentium PC (or compatible computer) with these characteristics:</p> <ul style="list-style-type: none"> • Clock speed — A minimum of 200MHZ. • Memory — A minimum of 64MB.
Hard disk drive	Sufficient disk space to hold the OpenEdge software you want to install. See the “Disk space requirements for OpenEdge installed products” section on page 3–8.
Disk space	Sufficient disk space on the drive where Windows is installed for Windows system files that OpenEdge copies there. Depending on which Windows system files are already installed on your system, OpenEdge can install up to 10MB in your Windows directory.
Java	See the “Java considerations” section on page 1–2 for detailed information about the Java requirements if you are installing any of these products: the OpenEdge® Enterprise RDBMS, OpenEdge® Workgroup RDBMS, or the OpenEdge® Personal RDBMS, or the components OpenEdge SQL Client Access.

Table 1–1: Windows system requirements to run OpenEdge*(2 of 2)*

Component	Requirement
OpenEdge installation media	<p>Progress® Software Corporation supports various media to access the OpenEdge product installation program:</p> <ul style="list-style-type: none"> • CD image — Requires a CD drive. • DVD — Requires a DVD player. • Electronic Software Distribution (ESD) download — Supports downloading software images from the Progress Download Center available at http://www.progress.com/esd; the Web site requires a valid account that your company must establish with Progress Software Corporation to access OpenEdge software products and updates.
Network protocol	Your system must support the TCP/IP and UDP protocols.
Web server	<p>Your Web server must support one of the following interfaces:</p> <ul style="list-style-type: none"> • Microsoft Web server (IIS) or ISAPI compatible — For example, Microsoft Internet Information Server (IIS), Versions 5.x and later. The IIS Web server ships with Windows platforms, but is not installed by default. This Web server supports an “in memory” messenger (ISAPI) and CGI messenger. • Sun Web server or NSAPI compatible — For example, a Sun Web server (formerly Netscape, iPlanet and SunOne Web Servers). This Web server supports an “in memory” messenger (NSAPI) and the CGI messenger. • CGI 1.1 — For example, Microsoft Internet Information Server (IIS), Versions 3.x and 4.x, or Apache. This Web server provides support for the Common Gateway Interface (CGI) messenger.
Web browser	<p>To use WebSpeed®, you must have a Web browser. Most contemporary browsers such as Mozilla, Opera, or Microsoft Internet Explorer, will work with WebSpeed.</p> <p>To run WebSpeed® Workshop, Progress Software Corporation recommends you use Microsoft Internet Explorer, Versions 5 or 6 only. The version of Internet Explorer you are using must support JScript Version 2.0.</p>

Server compatibility

If you run an OpenEdge multi-user system that includes older versions of OpenEdge (or Progress) products, make sure that your servers are compatible. The following sections detail OpenEdge RDBMS, supported Progress and OpenEdge DataServers, Progress® AppServer™, Transaction Server, AdminServer, and OpenEdge® Name Server™ compatibility.

OpenEdge RDBMS compatibility

A client for the Database Server can be any of the following:

- ABL client (character or GUI)
- WebSpeed Agent
- AppServer

Table 1–2 lists the client versions that are compatible with the database server versions. Progress Software Corporation does not support combinations not listed in the table.

Table 1–2: Client/Database server compatibility

Client version	Server version
OpenEdge 10.1A and 10.1B	<ul style="list-style-type: none">• Database server Progress Version 9• Database server OpenEdge 10.0B• Database server OpenEdge 10.1B
WebSpeed® Messenger 10.0B and 10.1B	<ul style="list-style-type: none">• Transaction Server 10.0B• Transaction Server 10.1B
WebSpeed Messenger 3.1E	<ul style="list-style-type: none">• Transaction Server 3.1E• OpenEdge Server 10.0B• OpenEdge Server 10.1B
WebSpeed Messenger 10.0B and 10.1B	Transaction Server 3.1E

Note: All releases support TCP/IP networking.

OpenEdge DataServer compatibility

A client for the DataServer can be any of the following:

- ABL client (character or GUI)
- WebSpeed Agent
- AppServer

Note the following compatibility rules.

Deployment rules supported by a DataServer broker

If you are using the DataServer brokering technology for an n-tier deployment, the OpenEdge versions of the client and the DataServer broker must be at the same maintenance level.

For example, an OpenEdge 10.1A client cannot connect to an OpenEdge 10.1B DataServer broker.

Development rules related to schema holder compatibility

Normal OpenEdge RDBMS database rules apply if the Schema Holder does not contain newer features or functionality that are not supported by the lesser release being used by the client.

For example, OpenEdge 10.1A and 10.1B clients can connect to an OpenEdge 10.1A-created schema holder being served in multi-user mode by an OpenEdge 10.1B RDBMS broker. However, if the 10.1A schema holder is updated using OpenEdge 10.1B and new features enabled (for instance BIGINT or 64Bit sequences) then the OpenEdge 10.1A client will no longer be supported, and you might experience runtime errors.

OpenEdge SQL Database Server

A client for the OpenEdge SQL Database Server can be either of the following:

- An OpenEdge SQL ODBC client.
- An OpenEdge SQL JDBC client.

Table 1–3 details client and server compatibility for an OpenEdge SQL Database Server. Progress Software Corporation does not support combinations not listed in the table.

Table 1–3: Client/OpenEdge SQL Database Server

Client version	Server version ¹
OpenEdge 10.0B	OpenEdge 10.0B and 10.1B
OpenEdge 10.1B	OpenEdge 10.1B
Progress 9.1E	Progress 9.1E

1. The administration of a server is associated with the server itself and therefore must be the same version. Therefore, the Database Administration Tools, including the Data Dictionary, Progress Explorer, and Admin Server that are used to administer the OpenEdge SQL Database must be the same version as those listed in the Server version column of the table.

AppServer compatibility

A client for the AppServer can be any of the following:

- ABL client
- Java client
- WebSpeed Agent
- AppServer
- Web Services Adapter
- OpenEdge Adapter for Sonic ESB®
- .NET client

[Table 1–4](#) details client and server compatibility for an AppServer. Progress Software Corporation does not support combinations not listed in the table. For .NET client-related details, see the “[.NET Client details](#)” section on page 1–8.

Note: The OpenEdge® NameServer also follows the compatibility rules listed in [Table 1–4](#).

Table 1–4: Client/AppServer compatibility

Client version	DataServer version
Progress 9.1E	OpenEdge 10.0B and 10.1B
OpenEdge 10.0B	OpenEdge 10.0B and 10.1B
OpenEdge 10.1A ¹	OpenEdge 10.0B and 10.1B

1. A 10.1B client can work with a 9.1E AppServer provided that the new functionality of the Version 10.1B, 10.0B or 10.0A client is not attempted with the 9.1E Server.

.NET Client details

OpenEdge 10.1B only supports Progress Version 9 AppServers, provided that the Version 10.1B clients only use Version 9 functionality. If a Version 10.1B client passes Version 9 temp tables, OpenEdge 10.0B and 10.1B will map them to ADO.NET data tables. ProDataSet functionality is not supported in this configuration.

[Table 1–5](#) details .NET client and server compatibility for an AppServer.

Table 1–5: .Net Client/AppServer compatibility

.Net client version	AppServer version
OpenEdge 10.0B	OpenEdge 10.0B and Progress 9.1E ¹
OpenEdge 10.1B	OpenEdge 10.0B and 10.1B, and Progress 9.1E ¹

1. Provided that only Progress Version 9 functionality is used.

Note: Progress Software Corporation does not support combinations not listed in the table.

Transaction server compatibility

The Transaction Server's client is the WebSpeed Messenger. [Table 1–6](#) details client and server compatibility for a Transaction Server.

Note: Progress Software Corporation does not support combinations not listed in the table.

Table 1–6: Transaction Server compatibility

WebSpeed Messenger client version	Transaction Server version ^{1 2}
OpenEdge 10.1B and 10.0B	OpenEdge 10.1B and 10.0B
WebSpeed 3.1E	WebSpeed 3.1E, and OpenEdge 10.1A and 10.0B
OpenEdge 10.1B	WebSpeed 3.1E
OpenEdge 10.0B	WebSpeed 3.1E

1. The administration of a server is associated with the server itself and therefore must be the same version. Also, the Administration Tools associated with the Transaction Server, including the Progress Explorer and the Admin Server that are used to administer the Transaction Server, must be the same version as those listed in the Server version column of the table.
2. A Transaction Server will work with a 10.1A NameServer. An OpenEdge 10.1A Transaction Server can work with a Progress 9.1E, OpenEdge 10.0B, or OpenEdge 10.0A NameServer, provided that you do not attempt to use the 10.1A NameServer new functionality with an earlier version of the NameServer.

Required third-party applications

The following sections describe the third-party applications that OpenEdge will install during the OpenEdge installation process if you do not already have the minimum required versions of these third-party applications on your system.

Products that require Microsoft .NET Framework

OpenEdge Release 10.1B supports a new process by which users are required to explicitly accept the License Agreement for the Microsoft .NET Framework prior to using the .NET Framework software.

Conditions that govern the display of the License Agreement

The following OpenEdge development products require the Microsoft .NET Framework to be installed to run:

- 4GL Development System
- OpenEdge® Development Server
- OpenEdge® Studio
- OpenEdge® Architect
- WebSpeed® Workshop

If the Microsoft .NET Framework is not currently installed on the system on which you plan to install OpenEdge Version 10.1B, and you are installing any of the previously listed deployment products, the OpenEdge Installation Utility will automatically launch the Microsoft .NET Framework installation once the OpenEdge Installation Program ends.

Note: If you plan to distribute a Silent installation that includes OpenEdge products that require Microsoft .NET Framework as part of the installation process, verify that the .NET Framework software is available on the system to which you are installing **before** you initiate the installation. Otherwise the Silent installation process will terminate.

Installing the DataDirect SQL ODBC drivers and the DataDirect ODBC branded drivers

If you are installing OpenEdge in a Windows operating system, you must have the following Microsoft Installers for this installation:

- The MDAC 2.6 Installer from Microsoft to install Microsoft Data Access Components (MDAC) in Windows operating systems. The OpenEdge Installation Utility contains the MDAC 2.6 Installer, which it automatically launches during installation. The MDAC 2.6 Installer is located at *OpenEdge-install-dir\odbc\install\mdac_typ.exe*. You can find more information about the MDAC at Microsoft's Web site at the following URL: <http://www.microsoft.com/data/prodinfo.htm>.
- The DCOM98 Installer from Microsoft to install the Distributed Component Object Model (DCOM). The OpenEdge Installation Utility installer contains the DCOM98 Installer, which it automatically launches during installation. The DCOM98 Installer is located at *OpenEdge-install-dir\odbc\install\dc98.exe*. You can find more information about DCOM98 at Microsoft's Web site by searching on DCOM98 at the following URL: <http://www.microsoft.com/>.

You must reboot your system after installing either the MDAC or the DCOM98 Installer.

DataDirect SQL ODBC drivers

In Release 10.1B, the SQL drivers have been upgraded from Version 5.1 5.2. The Installation Utility installs the DataDirect SQL ODBC driver files to the *installation-path\bin* directory. The Installation Utility also installs the files to %windir%\system32 for Windows operating systems. Table 1–7 lists the SQL driver files installed for the OpenEdge database.

Table 1–7: Driver files — Windows operating systems for SQL

Database	Driver files
OpenEdge SQL	pgodbc.lic pgicu22.dll pgoe1022.dll ¹ pgoe1022r.dll

1. Identifies the primary OpenEdge driver file. The pgoe1022.dll must be registered as a file data source name (DSN).

Installing the DataDirect branded ODBC drivers

The Installation Utility installs the DataDirect branded ODBC driver files to the *OpenEdge-install-dir\bin* directory. [Table 1–8](#) lists the DataDirect branded ODBC driver files installed with the OpenEdge® DataServer for ODBC.

Table 1–8: Driver files — Windows operating systems with OpenEdge DataServer for ODBC

Database	Driver files
All ODBC data sources	P1ASE21.DLL P1ASE21R.DLL P1DB221.DLL P1DB221R.DLL P1ICU21.DLL P1MSS21.DLL P1MSS21R.DLL IVP1.LIC
DB2 Common Server	P1DB221.d11 P1DB221r.d11
Microsoft SQL Server 2000 and later	P1MSSS21.d11 P1MSSS21r.d11
Sybase	P1ASE21.d11 P1ASE21r.d11

Note: Refer to the *odbcref.pdf* file installed in the *OpenEdge-install-dir\bin* subdirectory for information on how to configure data sources to connect to the different databases that OpenEdge supports.

Additional DataServer information for ODBC-related components

To use DataServer for ODBC, note the following:

- Progress Software Corporation recommends that you use DataDirect Version 5.1.
- You need the specific data-source software components and version numbers installed for the specific data sources you are using.

Table 1–9 presents the specific data-source software requirements.

Table 1–9: Data-source components and version numbers

If you are using the DataServer for ODBC product from this data source . . .	Then you must install . . .
DB2	One of the following software requirements: <ul style="list-style-type: none"> • DB2/MVS Version 3.1, 4.1, or 5.1. (These versions can include DDCS Multi-user Gateway Version 2.3 or later.) Also, DB2/MVS Versions 3.1 and 4.1 require IBM Client Access Enabler Version 2.11 or later. • DB2/NT Version 5.0 or 6.1. (Either version can also include DB/2 Connect Version 5 or later.) • DB2/6000 Version 5.0 or 6.1 (Either version can also include DB/2 Connect Version 5 or later.) • DB2/400 Version .5 operating system (Version 5.3).
DB2/400	DB2/400 Version 5.3.
Sybase	Sybase adaptive Server System 10 or System 11 for data-source server software and Open Client Library Version 11.1.

Platform Support for OpenEdge Management 3.1B

OpenEdge Management Release 3.1B is supported on these OpenEdge 10.1B Windows platforms: Windows 2000, Windows 2003, and Windows XP Professional.

Note: Support for OpenEdge Management 3.1B on UNIX/Linux platforms has been expanded. See the “[Expanded platform support for OpenEdge Management](#)” section on page 2–10.

For more information about OpenEdge Management 3.1B, see *OpenEdge Management: Installation and Configuration guide* on the PSDN Web site
<http://www.psdn.com/library/kbcategory.jspa?categoryID=129>.

UNIX Systems Installation Requirements

This chapter describes the requirements for installing OpenEdge Release 10.1B on a machine running a UNIX or Linux operating system. This chapter also identifies several other supported platforms on which you can install and run OpenEdge software, as outlined in the following sections:

- [UNIX system requirements](#)
- [ABL client/database server compatibility](#)
- [Supported platforms](#)
- [Licensing](#)

UNIX system requirements

This section describes the hardware and software requirements for running OpenEdge Release 10.1B on UNIX and Linux.

Ensuring you have the most up-to-date system requirements information

The system requirements provided in this chapter are current as of the publication date of this manual; however, requirements can change. To ensure that you have the most up-to-date information about system requirements, refer to *OpenEdge Release Notes* available with your installation medium and *OpenEdge 10 Platform & Product Availability Guide* on the Progress Software Corporation (PSC) Web site

<http://www.progress.com/products/lifecycle/index.ssp>, as needed.

Java requirements

Many OpenEdge products require the Java Runtime Environment (JRE), the Java Development Kit (JDK), or both of these components to use specific product functionality once the products are installed.

JDK component

The JDK contains the software and tools that developers need to compile, debug, and run applets and applications written using the Java programming language. The JDK software and documentation, typically included with each new release of an operating system, are available for download at the vendor's Web site. You need a JDK if you intend to develop Java stored procedures and triggers with the database, or to create Java proxies with the Progress® Open Client Toolkit.

Note: For details about the Release 10.1B supported platforms and specific Java requirements needed to support an OpenEdge installation on each platform, see the [“Supported platforms and Java components”](#) section on page 2–4.

Ensuring that the correct Java version is installed and recognized

The following conditions must exist to ensure that the correct Java version is properly installed on your machine and it is accurately recognized by the OpenEdge installation:

- Install the certified JDK to be used with OpenEdge Release 10.1B **before** you install OpenEdge.
- Ensure that the JDK is located in the `$PATH` environment variable to ensure that the OpenEdge installation can tailor the `java_env` file.
- The `$PATH` environment variable must point to the correct Java installation **before** you run the `proinst` utility. Otherwise, the system's default Java executable's version will be referenced from the `PATH`; the system's default is not necessarily the correct Java version for the OpenEdge installation.
- Ensure that the JDK is located in `$JAVAHOME/bin` environment variable so that the Installation Utility can find it. (The `JAVAHOME` `PATH` is the Java installation directory.)

If the correct Java software is not supplied with your installation medium, you must ensure that it is correctly installed on your system, according to the previously criteria.

OpenEdge Products that required the JRE

The JRE consists of the Java Virtual Machine, the Java Core Classes, and supporting files. The JRE is the run time part of the JDK and does not include a compiler, a debugger, or development tools. You must have the JRE if you intend to use one of the following:

- Progress Explorer
- WebSpeed Transaction Server
- OpenEdge® Application Server Basic
- OpenEdge® Application Server Enterprise
- Java application or applet
- OpenEdge® Adapter for SonicMQ®
- Web Services Adapter
- Secure AppServer Internet Adapter (AIA)

You must have the JRE to execute Java stored procedures and triggers from the database.

JDK and specific OpenEdge platforms

The JDK ships with OpenEdge Release 10.1B products that run on either the Solaris SPARC 32- or 64-bit platforms.

JRE and specific OpenEdge platforms

The JRE ships with OpenEdge Release 10.1B products that run on the following platforms:

- HP Tru64
- HP-UX (PA-RISC) (32-bit)
- HP-UX64 (PA-RISC) (64-bit)
- HP-UX (ITANIUM)
- Solaris SPARC (32-bit)
- Solaris SPARC (64-bit)

Note: If you are using the 32-bit JRE on a 64-bit machine, Java triggers and stored procedures are not supported.

Supported platforms and Java components

As mentioned earlier in this chapter, on the supported Sun Solaris platforms, the OpenEdge installation program also automatically installs the JDK when you install a product that requires the JDK.

The following list identifies other supported platforms for which you might need to install some required Java components:

- **HP-UX** — The installation program does not automatically install the JDK component if you install OpenEdge on this platform. If you want access to the JDK and do not already have it installed, you should install the required version of the JDK on your target system. For more information on Java requirements, see [Table 2–1](#).

Additionally, it might be necessary for you to adjust the default kernel parameter `max_thread_proc`. To determine whether the default kernel parameter is too low and should be modified, contact your system administrator.

- **AIX and Linux platforms** — The installation program does not automatically install the JRE or JDK component if you install an OpenEdge component on one of these platforms. You must install the required version of the JRE and/or the JDK (if not already installed) on your target system. For more information on Java requirements, see [Table 2–1](#).
- **OpenEdge SQL** — The installation program does not automatically install the JDK component when you install the OpenEdge RDBMS Enterprise Edition, the OpenEdge RDBMS Personal Edition, or the OpenEdge RDBMS Workgroup Edition with the OpenEdge SQL Client Access. If you intend to develop Java stored procedures and Java triggers for your database, you must install an OpenEdge development product. For information on writing Java stored procedures and triggers, see [OpenEdge Data Management: SQL Development](#).

Operating systems and JDK and JRE version details

[Table 2–1](#) lists operating systems, specifies the versions of JDK and JRE each supports, and provides a URL for more information about the JDK associated with a platform.

Table 2–1: JRE/JDK requirements by platform

(1 of 2)

Platform	Entry level JDK/JRE required	JDK location
Solaris SPARC (32-bit and 64-bit)	1.4.2_06 (JDK and JRE shipped with OpenEdge)	http://java.sun.com
HP-UX (PA-RISC) (32-bit and 64-bit)	1.4.2_06 (JRE ships with OpenEdge Release 10.1B)	http://www.hp.com/java
HP-UX ITANIUM (IA 64)	1.4.2_06	http://www.hp.com/java
Tru64 UNIX 5.1B(64 bit)	1.4.2_04	http://www.hp.com/java
RedHat Enterprise Linux 2.1	1.4.2_06 (Sun)	http://www.java.sun.com
SuSE Linux Enterprise Server 8	1.4.1_06 (Sun)	http://www.java.sun.com
IBM AIX 5.2 (32-bit and 64-bit)	1.4.2	http://www.ibm.com/java

Table 2–1: JRE/JDK requirements by platform (2 of 2)

Platform	Entry level JDK/JRE required	JDK location
UnixWare 7.1.4	1.4.2_06	http://www.sco.com/unixware
Linux PowerPC	1.4.2 SR3	http://www-128.ibm.com

► To determine what version of Java you currently have on your operating system, type **java -version** at the command line.

Note: On some platforms, multiple versions of Java may be available.

You can change your PATH variable to reference a different version, if needed. In the case of the Open Client Toolkit, other Java versions (including versions from other vendors) can also be used.

When SUN provides a JDK and JRE for a certain platform, Progress Software Corporation (PSC) includes the JDK and JRE in its distribution. For other systems, you must obtain the JDK from the system’s operating system vendor. Contact your operating system vendor for assistance in obtaining the JDK.

Ensuring you have the most up-to-date Java requirements information

The Java requirements and support strategies presented in the previous section are the most current as of the time this book was published. For details about any changes to this information, see *OpenEdge 10 Platform & Product Availability Guide* on the PCS Web site <http://www.progress.com/products/lifecycle/index.ssp>.

Requirements for building ABL applications

Developing a character-based ABL application requires one of the following:

- A character terminal attached to a computer.
- A PC that is connected to a host computer.

Requirements for running OpenEdge applications

Table 2–2 lists the minimum requirements for running OpenEdge applications.

Table 2–2: Minimum requirements for running OpenEdge applications

Component	Requirement
Terminal	A character terminal attached to a host computer. Note: OpenEdge does not support spacetaking terminals unless the terminal has a firmware setup option to change it to nonspacetaking mode.
Libraries	Networking libraries must be installed on your machine. Multi-user OpenEdge configurations connect UNIX-to-UNIX through the OpenEdge-supported network protocol TCP/IP. You can also connect to a UNIX server from a Windows client through TCP/IP.
JDK	Installed JDK components. See Table 2–1 for the current version releases.

Product and application dependencies

Additional requirements might exist, depending on the applications you plan to run and/or the OpenEdge products you plan to install. For example, you might need any or all of the following elements:

- **Web server** — To run WebSpeed. The Web server types supported include Microsoft IIS Web server or ISAPI-compatible, Sun Web server or NSAPI-compatible, or CGI-compatible.
- **Java servlet engine** — To run Web Services.
- **JRE/JVM** — To run Java applications.

File descriptors

You must allocate sufficient file descriptors to handle all the WebSpeed Agents your configuration uses. Set the file descriptors to 1024 by entering the following command:

```
ulimit -n 1024
```

ABL client/database server compatibility

If you run an OpenEdge multi-user system with more than one OpenEdge release, make sure that your ABL client and database server are compatible.

Table 2-3 lists the ABL client versions that are compatible with the OpenEdge database server versions. (Note that all releases support TCP/IP networking.)

Table 2-3: Client/server compatibility

ABL Client	Database server
OpenEdge Release 10.1B	OpenEdge Release 10.1B and 10.0B
OpenEdge 10.0B	OpenEdge Release 10.1B and 10.0B
OpenEdge 10.X	Progress Release 9.1.E

Supported platforms

This section describes the platforms supported in OpenEdge 10.1B, and provides additional details about specific platforms and platform-related features. Refer to the hard-copy and online *OpenEdge Release Notes* for additional requirements.

Table 2-4 lists the platforms supported with this release and the minimum operating system level supported.

Table 2-4: Supported platforms (1 of 2)

Platform	Entry level operating system
HP-UX (PA-RISC) (32- and 64-bit)	HP-UX 11.11
HP-UX ITANIUM (IA 64)	HP-UX 11i v2 (11.23)
HPTru64	HP Tru64 UNIX 5.1B
IBM AIX (32-bit)	IBM AIX 5. 2
IBM AIX (64-bit)	IBM AIX 5.2

Table 2–4: Supported platforms

(2 of 2)

Platform	Entry level operating system
Solaris SPARC (32- and 64-bit)	Solaris 9
Linux x86 (32-bit)	Red Hat Enterprise Server 2.1 SuSE Enterprise Server 8.0
Linux (x86) (ADM64, EMG4T)	SuSE Enterprise Server, 9.0 Red Hat Enterprise 3.0
Sco UnixWare	UnixWare 7.1.4 ¹
Linux PowerPC	RedHat Enterprise Server 4.0

1. UnixWare 7.1.4 supports multi-threaded capabilities available in OpenEdge SQL and in the OpenEdge database utilities. The addition of these capabilities to UnixWare completes the suite of OpenEdge-supported platforms that allow users to employ multi-threaded features.

This is the most up-to-date information on supported platforms. For the most recent changes to the list of supported platforms, see *OpenEdge 10 Platform & Product Availability Guide* on the PSC Web site <http://www.progress.com/products/lifecycle/index.ssp>.

Support for Linux PowerPC

Release 10.1B supports Linux PowerPC, a 64-bit platform on the Linux Operating System. The minimum operating system level for this platform is Red Hat Enterprise 4.0. The minimum Java level for this platform is 1.4.2 SR3.

The PowerPC platform enables the delivery of a 64-bit implementation of OpenEdge products. The implementation includes all Unix/Linux OpenEdge functionality with the exception of the following deliverables or functionality:

- Web Services Out (Web Services Producer)
- XML
- Failover Clusters
- OpenEdge Adapter for Sonic MQ
- OpenEdge Adapter for Sonic ESB
- ODBC Client for OpenEdge SQL Databases

Support for Failover Clusters

OpenEdge Release 10.1A (or earlier) support Failover Clusters on the following platforms: AIX 32, HP UX 32, HP UX 64, Solaris 32, and Solaris 64 (as well as Windows).

Starting with Release 10.1B, the following new platforms also support Failover Clusters:

- HP Itanium (IA64) — The clustering environment on HP Itanium (IA64) is identical to the HP UX (PA-RISC) 32- and 64-bit Failover Clusters environment.
- AIX 64-bit platforms — The cluster software for AIX 32 and AIX 64 is now HACMP 5.3.

In all of these platforms, Failover Clusters is installed as a component of the OpenEdge Enterprise RDBMS.

Expanded platform support for OpenEdge Management

In Release 10.1B, support for OpenEdge Management Release 3.1B has been expanded to include the following supported 64-bit platforms:

- HP-UX ITANIUM (IA 64)
- Linux x86 (AMD, EM64T)
- Linux PowerPC

With the exception of Sco Unixware, the latest 64-bit platform support completes the OpenEdge 10.1B suite of supported UNIX/Linux platforms on which OpenEdge Management 3.1B is now supported:

- HP-UX (PA-RISC) (32- and 64-bit)
- HPTru64
- IBM AIX (32- and 64-bit)

- Solaris SPARC (32- and 64-bit)
- Linux x86 (32-bit)

Note: OpenEdge Management Release 3.1B is also supported on these OpenEdge 10.1B Windows platforms: Windows 2000, Windows 2003, and Windows XP Professional for Intel 86 architecture.

For more information about OpenEdge Management 3.1B, see *OpenEdge Management: Installation and Configuration guide* on the PSDN Web site
<http://www.psdn.com/library/kbcategory.jspa?categoryID=129>.

JDBC Type 4 driver support on 64-bit platforms

Release 10.1A introduced the all-Java based JDBC Type 4 driver whose primary function is to allow Java-based applications native access to the OpenEdge database. In Release 10.1B, the JDBC Type 4 driver is now supported on additional OpenEdge-supported platforms, including Solaris SPARC (32- and 64-bit), HP-UX (PA-RISC) (32- and 64-bit), and HP-UX Itanium (IA 64).

When the JDBC Type 4 driver is installed on any 64-bit platforms—HP Itanium (IA64), Linux 64, and Linux PowerPC—the driver must be used in concert with the JDK/JRE 1.4 supported environments.

For more information about JDBC Type 4 drivers, see *OpenEdge Data Management: SQL Development*.

Licensing

When installing OpenEdge, the installation utility prompts you to enter product information from your *License Addendum*. The installation utility records the license information in the OpenEdge configuration file, `progress.cfg`.

The following syntax shows how to use the Show Configuration (SHOWCFG) utility to display the product license information for each OpenEdge product installed on your system:

```
OpenEdge-install-dir/bin/showcfg OpenEdge-install-dir/progress.cfg
```

Note: When you install a client networking license, the ADM/ADM2 directories are not installed in the *OpenEdge-install-dir/tty* directory. This r-code is considered part of your application and should be deployed as a module of your application.

For more information about licensing, see [Chapter 6, “Administration Utilities.”](#)

OpenEdge Installation Prerequisites

This chapter presents prerequisite information and some preliminary tasks to perform before you install OpenEdge software on any of the supported platforms, as described in the following sections:

- [Tasks overview](#)
- [Gathering information to plan your installation](#)
- [Determining your installation method](#)
- [Determining the type of installation](#)
- [Shared Network Installation Utility](#)
- [Determining the type of installation](#)
- [Disk space requirements for OpenEdge installed products](#)
- [Windows-specific installation considerations](#)
- [UNIX-specific installation considerations](#)
- [WebSpeed configuration choices](#)
- [Installing or viewing product documentation and samples](#)

Tasks overview

Complete the following preinstallation tasks before starting your OpenEdge installation:

- ☐ Gather product-related documents to make informed decisions about your OpenEdge installation before you begin. See the [“Gathering information to plan your installation”](#) section on page 3–3.
- ☐ Determine the installation method you plan to perform: Online or Silent. See the [“Determining your installation method”](#) section on page 3–5.
- ☐ Determine the type of installation you plan to perform: complete or custom. See the [“Determining the type of installation”](#) section on page 3–6.
- ☐ Review disk space requirements to accommodate OpenEdge installed products. See the [“Disk space requirements for OpenEdge installed products”](#) section on page 3–8.
- ☐ Review other required applications. See the [“Windows-specific installation considerations”](#) section on page 3–12. See the [“Windows-specific installation considerations”](#) section on page 3–12.
- ☐ Save or upgrade an existing OpenEdge or Progress installation (if applicable). See the [“Saving an existing OpenEdge or Progress installation in Windows-based platforms”](#) section on page 3–14 and the [“Upgrading an existing OpenEdge or Progress installation on UNIX platforms”](#) section on page 3–24.

Note: Tasks are considered common to all supported platforms unless otherwise specified.

Gathering information to plan your installation

Table 3–1 identifies and briefly defines the various documents that Progress Software Corporation (PSC) provides to help you plan your installation.

Note: The OpenEdge 10.1B Installation Utility in Windows and on UNIX is available from these sources: CD, DVD, and Electronic Software Download (ESD). In some instances, the specific installation medium you use determines a document’s delivery method.

Table 3–1: Preinstallation documentation resources

(1 of 2)

For this task . . .	Reference the . . .	Which is available as . . .
Determine and record product installation choices,	Appendix A, “Preinstallation Checklist for Windows” or Appendix B, “Preinstallation Checklist for UNIX”	<ul style="list-style-type: none"> • An appendix in this guide. • A hardcopy document in your OpenEdge product box. • A PDF-formatted document available from the Documentation link of the Progress Web site.¹ • A PDF-formatted document on the OpenEdge Documentation and Sample CD.² • A document accessible from the root directory of the CD or DVD installation media.³ • A document downloadable from the Progress Download Center.⁴
Identify serial numbers and product control codes associated with products to install,	<i>License Addendum</i>	<ul style="list-style-type: none"> • A hardcopy document in your OpenEdge product box. • A document downloadable from the Progress Download Center.⁴

Table 3–1: Preinstallation documentation resources

(2 of 2)

For this task . . .	Reference the . . .	Which is available as . . .
Identify product-related details that might not be in the OpenEdge 10.1B product documentation,	<i>OpenEdge Release Notes</i>	<ul style="list-style-type: none"> • A paper document in your OpenEdge product box. • A document downloadable from the Progress Download Center.⁴ • As an online document accessible from specific menu options on UNIX installation dialog boxes.
Reference each product's components and subcomponents, as needed,	Chapter 12, "OpenEdge Installation Products and Components in Windows" or Chapter 13, "OpenEdge Installation Products and Components on UNIX."	A chapter in this guide.
Complete the checklist to prepare for the installation and perform each procedure online during the installation process,	Windows OpenEdge Installation online help and UNIX OpenEdge Installation online help.	<ul style="list-style-type: none"> • A specific document accessible from either the root directory of the CD or DVD installation media.⁵ • A specific document downloadable from the Progress Download Center. • During either the Windows or UNIX installation process. Choose Help or another specific Help control on an installation dialog box.

1. The Documentation link on the PSDN Web site is located at <http://www.psdn.com/library/kbcategory.jspa?categoryID=129>.
2. The OpenEdge Documentation and Sample CD is shipped in the OpenEdge product box. For details about the documentation and samples you can install in Windows and on UNIX, see the "Installing or viewing product documentation and samples" section on page 3–28.
3. The checklist documents file names are "Windows_Preinstall_Checklist.pdf" and "Unix_and_Linux_Preinstall_Checklist.pdf."
4. The Progress Download Center is located at <http://www.progress.com/esd>. You must have a valid user name and password to download products from this site. Contact a Progress Customer Service Representative to set up your Download Center account.
5. The Windows online help document's filename is `inst_hlp.chm`. The UNIX online help document's filename is `inst_hlp.txt`.

Determining your installation method

You can install OpenEdge Release 10.1B products in Windows or on UNIX using either of these methods:

- **Online, interactive installation** — This method prompts you to make installation choices online and record your input in dialog boxes. The dialog boxes appear programmatically as determined by the products you identify to install and the type of install you choose to perform. After you complete the Installation Utility, the Setup Utility initializes your choices, enabling you to use the products after the installation.

For details about loading your installation medium and initiating either a Windows installation or a UNIX installation, see:

- [Chapter 4, “Performing an OpenEdge Installation in Windows”](#)
- [Chapter 5, “Performing an OpenEdge Installation on UNIX”](#)

Note: Online help, provided with each online platform’s Installation Utility and accessible from **Help** or a specific **Help** control, presents the detailed procedures to complete each installation dialog box.

- **Silent Installation Utility** — The silent, or batch mode, installation does not prompt you to interactively enter your installation choices. A silent installation reads your installation values and settings as recorded in a response file. Using specific commands, you initiate your response file to run without user involvement. A silent installation supports either a Complete or Custom installation.

For details about running silent or batch installations by creating a response file, see:

- The [“OpenEdge Silent installation overview”](#) section on page 4–31 in Windows platforms.
- The [“OpenEdge Silent installation overview”](#) section on page 5–13 on UNIX platforms.

Determining the type of installation

Your OpenEdge installation process depends on what products, components, and subcomponents you choose to install and the type of installation you plan to perform. [Table 3–2](#) summarizes the installation options.

Table 3–2: Installation options

Installation option	Purpose
Complete	Automatically installs all mandatory, recommended, and optional components and subcomponents of the OpenEdge products you are installing.
Custom	<p>Installs all mandatory product components and subcomponents, but allows you to selectively install the recommended and optional components and subcomponents on a product-by-product basis.</p> <p>A custom installation provides more advanced users, for whom this method is recommended, flexibility to distribute OpenEdge components on different machines, select product components to suit their business needs, and work around issues such as disk space limitations.</p> <p>Note: When customizing an install, Progress Software Corporation recommends that you consider removing only optional components and subcomponents. Removing recommended products might negatively affect a product’s functionality.</p>

How product selection can affect your installation tasks

Some OpenEdge products that you install have additional installation dependencies, prompting you to perform additional set up or installation tasks as part of the OpenEdge installation process. The OpenEdge Installation Program automatically determines certain product dependencies and guides you through the completion of these tasks. The following list identifies some of the ways the products you install affect your installation experience:

- If you are performing a Windows installation, OpenEdge 10.1B development products require Microsoft .NET Framework installed on the machine to which you are installing OpenEdge. If it is not already installed, you will be prompted to accept the software's installation at the conclusion of the Installation Program. See the [“Required third-party applications”](#) section on page 1–10 and the [Appendix A, “Preinstallation Checklist for Windows”](#) for details.
- To complete an OpenEdge installation that includes Progress Dynamics®, the Installation Program automatically launches the Progress Dynamics Configuration Utility (DCU). For details about the DCU utility procedure, see the [“Completing the DCU wizard”](#) section on page 4–7.

To review a complete list of all OpenEdge products, and the components and subcomponents that comprise each, see [Chapter 12, “OpenEdge Installation Products and Components in Windows”](#) or [Chapter 13, “OpenEdge Installation Products and Components on UNIX”](#) to review product component and subcomponent details.

Shared Network Installation Utility

In a Windows-based platform, you can provide multiple-client access to a single copy of OpenEdge that is installed on a network-accessible drive (server). To initiate this shared installation arrangement, you must choose to install the Shared Network Installation component (NetSetup) during the installation process. Then, using the NetSetup deployment option suitable to your site's needs, you can install the NetSetup whether you are performing a Complete or a Custom installation.

For more information about sharing an OpenEdge installation between a server and clients, see [Chapter 4, “Performing an OpenEdge Installation in Windows.”](#)

Disk space requirements for OpenEdge installed products

This section provides approximate file size information that each OpenEdge product requires on disk. This information will help you to allocate sufficient disk space for your installed OpenEdge products. This section addresses presents disk requirements for:

- [OpenEdge products installed in Windows.](#)
- [OpenEdge products installed on UNIX.](#)

Keep in mind that the actual space requirements can vary on either platform, depending on several factors, including considerations such as:

- The block size of your machine.
- The presence or absence of a JRE/JDK as part of the product.
- The bit size of your product executables. For example, 64-bit executables are larger than 32-bit executables.

OpenEdge products installed in Windows

[Table 3–3](#) identifies the approximate file size that each OpenEdge product requires based on the following criteria:

- Disk space is presented in MB blocks.
- A complete installation that includes support for one installed language.

Table 3–3: Disk space requirements for OpenEdge installed products in Windows
(1 of 3)

In Windows, this installed OpenEdge Product . . .	Requires the following number of MB blocks of disk space . . .
OpenEdge RDBMS Personal Edition	377
OpenEdge RDBMS Workgroup Edition	377
OpenEdge RDBMS Enterprise Edition	377
OpenEdge® DataServer for Oracle	321

Table 3–3: Disk space requirements for OpenEdge installed products in Windows (2 of 3)

In Windows, this installed OpenEdge Product . . .	Requires the following number of MB blocks of disk space . . .
OpenEdge® DataServer for ODBC	358
OpenEdge® DataServer for MS SQLServer	344
OpenEdge Development Server	748
OpenEdge Application Server — Basic	366
OpenEdge Application Server — Enterprise	392
OpenEdge Studio	881
OpenEdge Adapter Sonic ESB	152
OpenEdge SQL Client Access	146
Progress® Client Networking	346
Progress® Translation Manager	135
OpenEdge® Replication	139
OpenEdge® Replication Plus	139
4GL Development System	528
Progress® Visual Translator	345
Progress® Query/RESULTS	332
WebSpeed Workshop	750
WebSpeed Messenger	566
OpenEdge NameServer	146
OpenEdge® NameServer Load Balancing	103
OpenEdge Architect	890
AppServer Internet Adapter	55.5

Table 3–3: Disk space requirements for OpenEdge installed products in Windows (3 of 3)

In Windows, this installed OpenEdge Product . . .	Requires the following number of MB blocks of disk space . . .
Web Services Adapter	29.4
All OpenEdge Products (as identified in this table)	994

OpenEdge products installed on UNIX

Table 3–4 identifies the approximate file size that each OpenEdge product requires when installed on the following platforms:

- **Category one platforms** — Refers to these supported platforms: IBM AIX 5.2 (32-bit and 64-bit), Linux x86(32-bit and 64-bit), LinuxPowerPC, and Sco Unixware.
- **Category two platforms** — Refers to these supported platforms: HP-UX (PA-RISC) (32-bit and 64-bit), Solaris SPARC (32-bit and 64-bit), and Tru64.
- **Category three platform** — HP-UX Itanium (IA 64).

The approximate file size that each OpenEdge product requires is based on the following criteria:

- Disk space is presented in MB blocks.
- A complete installation that includes support for one installed language.

Table 3–4: Disk space requirements for OpenEdge products categorized by platforms*(1 of 2)*

Installed OpenEdge product	IBM AIX 5.2 (32-bit, 64-bit), Linux x86 (32-bit, 64-bit), Linux PowerPC, and Sco UnixWare¹	HP-UX (PA-RISC) (32-bit, 64-bit), Solaris SPARC (32-bit, 64-bit), and Tru64¹	HP-UX ITANIUM (IA 64)¹
OpenEdge RDBMS Personal Edition	419	463	772
OpenEdge RDBMS Edition	419	462	771
OpenEdge RDBMS Enterprise Edition	419	463	772
OpenEdge DataServer for Oracle	294	339	653
OpenEdge Development Server	629	683	1001
OpenEdge Application Server — Basic	410	453	756
OpenEdge Application Server — Enterprise	421	465	763
OpenEdge Adapter Sonic ESB	164	195	—
OpenEdge SQL Client Access	190	228	429
Client Networking	361	405	687
OpenEdge Replication	161	195	405
OpenEdge Replication Plus	161	195	404
4GL Development System	429	484	780
Progress Query/RESULTS	305	347	647
WebSpeed Messenger	60	58	93
OpenEdge NameServer	133	171	360
NameServer Load Balance	96	134	319
AppServer Internet Adapter	58	56	91

Table 3–4: Disk space requirements for OpenEdge products categorized by platforms

(2 of 2)

Installed OpenEdge product	IBM AIX 5.2 (32-bit, 64-bit), Linux x86 (32-bit, 64-bit), Linux PowerPC, and Sco UnixWare ¹	HP-UX (PA-RISC) (32-bit, 64-bit), Solaris SPARC (32-bit, 64-bit), and Tru64 ¹	HP-UX ITANIUM (IA 64) ¹
Web Services Adapter	31	29	56
All OpenEdge Products Installed (as identified in this table)	670	711	1068

1. Numbers represent averages that are within 10% utilization.

Windows-specific installation considerations

This section identifies the following Windows installation considerations:

- [OpenEdge working directory reminder.](#)
- [Ensuring .dll and .ocx files in the Windows directory are read-only.](#)
- [Software required to run OpenEdge products or components.](#)
- [Saving an existing OpenEdge or Progress installation in Windows-based platforms.](#)
- [Reviewing the Windows platform installation directory structure.](#)
- [Integrating OpenEdge with Windows Explorer.](#)
- [Installing or viewing product documentation and samples.](#)

OpenEdge working directory reminder

A *working directory* is a directory that contains your OpenEdge database and application files. The Installation Utility prompted you to create a working directory from which to run OpenEdge.

Caution: Never run OpenEdge products from the directory in which you installed them. If you do, you could damage the OpenEdge software files.

Ensuring .dll and .ocx files in the Windows directory are read-only

Before you install any OpenEdge products in Windows 2000, Windows 2003, or Windows XP Professional system, check the directory C:\Winnt\system32 or C:\Windows\system32 to see whether any of the .dll or .ocx files have the read-only bit set. If any .dll or .ocx files in this directory are read-only, you must reset them before installing OpenEdge. If you try to install OpenEdge with one or more related .dll or .ocx files set to read-only, OpenEdge will generate a dialog box to inform you that you must reset the .dll bit or .ocx bit and reinstall OpenEdge.

Software required to run OpenEdge products or components

Some OpenEdge products and/or components depend on the presence of other software or software elements to run as designed. These elements might be required either before you perform an OpenEdge installation or concurrent with the OpenEdge install you perform.

Microsoft Internet Explorer

If you are installing a product that contains Progress Explorer or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system to use the graphical administrative tools. If you do not have MS IE Version 4.01 or later, you will receive a warning message during the installation that tells you to install MS IE Version 4.01 or later. You can obtain information about acquiring or upgrading to MS IE Version 4.01 or later from the Microsoft Web site. You can continue with the installation after viewing this message, but neither ProxyGen nor the Progress Explorer graphical administrative tools will be functional.

Open Client Toolkit component

If you plan to install a product that contains ProxyGen, you might need to install and configure additional tools to allow the Open Client Proxy Generator (ProxyGen) to build proxies. For more information, see the chapter on configuration and deployment in *OpenEdge Development: Open Client Introduction and Programming*.

OpenEdge SQL

The installation program does not automatically install the JDK component when you install any of these products: the OpenEdge Enterprise RDBMS, the OpenEdge Workgroup RDBMS, or the OpenEdge Workgroup RDBMS with the OpenEdge SQL Client Access. If you intend to develop Java stored procedures and Java triggers for your database, you must install an OpenEdge development product such as OpenEdge Studio. For information on writing Java stored procedures and triggers, see *OpenEdge Data Management: SQL Development* and *OpenEdge Data Management: SQL Reference*.

OpenEdge SQL ODBC and JDBC Clients

The OpenEdge SQL ODBC and JDBC Clients are components of the OpenEdge Personal RDBMS, Workgroup RDBMS, and Enterprise RDBMS products. They can be downloaded, using the OpenEdge SQL Client Access product, from the Progress Software Corporation Web site at <http://www.progress.com/esd>.

Saving an existing OpenEdge or Progress installation in Windows-based platforms

If you have an existing OpenEdge or Progress installation, you might want to save certain pieces of it to make configuring your new installation simpler. If you want to continue using any templates, customized procedure or code files, a `progress.ini` file, you must copy them to another location before you begin a new installation.

Before you begin

Progress Software Corporation strongly recommends that you thoroughly examine and review your existing installation before you make any changes. The tasks to plan and save a current installation will vary, depending on several factors. Consult the following sources to help you determine your specific activities to save your existing installation. Also, review the general steps as noted in this section.

Resources to help you plan and save your current installation

To help plan and implement the tasks required to save your current installation, consult the following online resources for OpenEdge database documentation and appropriate white papers available on the:

- PSC Web site at <http://www.progress.com>.
- Progress Knowledge Center at <http://progress.atgnow.com/esprogress>.
- Progress Software Developers Network (PSDN) at <http://www.psdn.com/library/kbcategory.jspa?categoryID=129>.

In addition to the online resources, contact Progress Technical Support for help in the tasks associated with saving an existing OpenEdge or Progress installation.

Note: Do not install different versions of OpenEdge into the same *OpenEdge-install-dir* directory.



To save an existing installation:

1. If you have any templates that you want to continue using, copy them to another location before installing OpenEdge. If you have information in a `progress.ini` file that you want to continue using, copy the current `progress.ini` file to a directory other than where you are installing OpenEdge.
2. If you have any customized procedure or code files in the directory where you are installing OpenEdge, copy them into a different directory.

For more information on saving previous versions of your `progress.ini`, customized procedures, or code files, see the [“Performing an OpenEdge Installation in Windows”](#) section on page 4–1.

3. If you have OpenEdge installed and the `PROMSGS` environment variable is set on the **Environment** tab of the **System** settings in the Windows 2000 Control Panel, you must remove the `PROMSGS` environment variable before installing OpenEdge. If `PROMSGS` points to an old or nonexistent `PROMSGS` file, the InstallShield utility will not write all the necessary data to the Windows registry.

Caution: You must perform [Step 3](#) as described if you current installation meets the criteria defined. Otherwise, you will have unpredictable and undesirable results.

4. Truncate the before-image (`.bi`) file using the `PROUTIL TRUNCATE BI` utility and back up your existing database using the `PROBKUP` utility. For more information on these utilities, see [OpenEdge Data Management: Database Administration](#).

OpenEdge requires that your databases be converted to a multi-volume structure. If you were using single-volume databases with Progress Version 8 or Version 9, you must convert your OpenEdge databases to a multi-volume structure before converting the databases to OpenEdge. (Note that you must truncate your BI file before you convert it. If you plan to replace your current Progress Version files with OpenEdge 10, complete this step before you perform the installation to avoid erasing your current Progress Version files.)

Caution: This conversion task involves many steps and requires that you plan each of them. To plan your steps, consult these resources as noted in the [“Resources to help you plan and save your current installation”](#) section on page 3–15.

Existing JavaSoft (InstallShield) JDK

If the required version of the Java Soft (InstallShield) JDK has been installed on your system prior to the OpenEdge Release 10.1B installation and you want to use this pre-existing JDK utility, you can. However, you must first complete the OpenEdge installation and then, as a postinstallation task, edit files tailored by the install to ensure that they point to this pre-existing JDK. Contact Progress Technical Support for assistance to perform this task.

OpenEdge automatic save of Properties files

OpenEdge automatically makes copies of your `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences` files and places them in another directory, `*.sav`. The new installation automatically upgrades the files in the `install-path\properties` directory. However, after you have finished your new installation, you must replace the newly installed versions of these files with these copies. When you start the AdminServer, your older files will be updated to match the current standards for these files. For information about the procedure to uninstall an existing OpenEdge product and instantiating the properties files, see [Chapter 5, “Performing an OpenEdge Installation on UNIX.”](#)

When you uninstall an existing OpenEdge product, the process copies `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences`, the three files in the `install-path\properties` directory, to `%TEMP%`. After installing a new OpenEdge product, you can manually copy back the files from `%TEMP%`.

Reviewing the Windows platform installation directory structure

The OpenEdge installation PATH contains configuration files and several subdirectories. The installation PATH directory contains the OpenEdge executables, several procedure (`.p`) files, and other related files and subdirectories. The default OpenEdge installation PATH is `C:\Progress\OpenEdge`. However, during the OpenEdge Installation, you can choose a different location into which to install.

References to the Windows installation directory in this guide

Throughout this guide, the installation PATH is referred to as either of the following:

- **DLC** — The DLC variable in Windows, %DLC%, is automatically set to your OpenEdge installation PATH. Historically, it has been a convenient way to refer to the location in which you have installed OpenEdge.

Note that the %DLC% variable is set in the various command scripts and in the registry; the variable is not, and should not, be set at the system level. For information about the %DLC% environment variable, see [Chapter 7, “Working in the OpenEdge Environment in Windows.”](#)

- **OpenEdge-install-dir** — A more explicit phrase to refer to the directory location to which your OpenEdge installation PATH. The Windows environment variable DLC is also used to create this location; the use of the phrase *OpenEdge-install-dir* is intended to be more self-explanatory than is the reference %DLC%.

[Table 3–5](#) describes a directory tree of the OpenEdge subdirectories.

Table 3–5: OpenEdge-install-dir (%DLC%) directory structure (1 of 3)

Directory name	Description
auditing	Contains object (.r), development, and environment (ADM2) files for the Audit Policy Maintenance product.
bin	Contains the executable files for OpenEdge, such as PRODB. It also contains batch files and system executables.
certs	Contains the public keys of the Certificate Authorities (CAs) used by OpenEdge clients to perform server-side certificate validation when communicating with secure Web servers using HTTPS.
dotnet	Contains support files to develop and deploy the .NET client.
esbadapter	Contains the configuration and support code for the OpenEdge Adapter for Sonic ESB.
gui	Contains object (.r), development, and environment (ADE) files for the OpenEdge graphical tools. These tools are compiled to run in graphical mode in Windows 2000, Windows 2003, and Windows XP Professional; they cannot run in a character environment.
include	Contains C and C++ header files.

Table 3–5: OpenEdge-install-dir (%DLC%) directory structure (2 of 3)

Directory name	Description
install	Contains Java tailoring classes that only the Installation Utility uses. Also contains the automatically generated response.ini file used in an OpenEdge Silent installation.
java	Includes the Java files and executables necessary for running OpenEdge products.
javahelp	Contains .jar files for the OpenEdge Application Debugger.
jdk	Contains the Java Development Kit files and executables necessary for running OpenEdge products.
jms	Contains files to support client deployment of java messaging.
jre	Contains the Java Runtime Environment files and executables necessary for running OpenEdge products.
keys	Contains encrypted RSA Private Key and Certificate file information.
lib	Contains shared objects necessary for running OpenEdge executables.
licenses	Contains license and copyright information related to HTTP Client, Open SSL toolkit, Perl, and w3c IPR software notice.
netsetup	Contains files for the Shared Network Installation Utility.
odbc	Includes files to support ODBC.
oebuild	Includes files that the OEBUILD utility uses when creating custom executables.
oeide	Contains the Eclipse environment, the plugins that comprise the OpenEdge Architect product, and other related files.
ora	Contains files to support the DataServer for ORACLE.
perl	Contains files to support the use of the Perl scripting language.
proedit	Contains files to support the advanced editing features.
prohelp	Includes the online help and other necessary files for OpenEdge.
prokey32	Contains files for international keyboard support for the 32-bit Windows Character Client.

Table 3–5: OpenEdge-install-dir (%DLC%) directory structure *(3 of 3)*

Directory name	Description
prolang	Contains the national language support directories.
properties	Contains property files that manage the configuration of OpenEdge services, such as WebSpeed, the NameServer, and the AppServer.
scripts	Can contain files related to the Failover Cluster component.
servlets	Identifies the default location of the AppServer Internet Adapter (AIA) and Web Services Adapter (WSA) servlet containers. These containers include web definitions. ¹
sonic	Contains files that support the Sonic client and container.
sports	Includes the schema triggers and supplier information for each sample database.
sports2000trgs	Includes the schema triggers for the Sports2000 database.
src	Contains source files for OpenEdge ADE tools, such as the WebSpeed, Data Dictionary, Procedure Editor, and Sample Applications.
templates	Can contain files related to the Failover Cluster component.
toolkit	Includes files that help in deploying and encrypting your applications.
tty	Includes mostly object (.r) files for character-mode OpenEdge.
ubqmanager	Includes files used by the AppServer exclusively. Do not modify these files.
wcadd	Contains Web Client CD images that include, among other files, the setup.exe to install the Web Client.
webinstall	Contains several WebSpeed-related files, including samples, scripts, and help files.
webspeed	Supports WebSpeed Workshop files such as samples, scripts, and help, that reside on the Web server.

1. Refer to your OpenEdge product documentation for details about configuring WSA and AIA.

Integrating OpenEdge with Windows Explorer

In Windows platforms, Microsoft allows applications to integrate certain features with Windows Explorer. Among those features, OpenEdge supports defined icons, shortcut menus, and property sheets for several of its file types. You can now easily perform an action on a file or view detailed information about a file from Windows Explorer. This section presents:

- [OpenEdge file types](#)
- [Icons](#)
- [Shortcut menus](#)
- [Properties](#)

OpenEdge file types

OpenEdge supports defined icons and shortcut menus for the following file types:

- Procedure source code file (.p)
- Window procedure source code file (.w)
- Include file (.i)
- Parameter file (.pf)
- Configuration file (.cfg)
- Database file (.db)

OpenEdge also supports specific property information for these file types:

- Compiled procedure code file (.r)
- Database file (.db)

This information is stored in the registry, separately from your OpenEdge settings.

If another application has already registered a file extension that OpenEdge uses, the Installation Utility asks if you want to overwrite the information for that file extension. If you choose no, OpenEdge does not display the icon, shortcut menu options, or properties information for that file type. If you choose yes, OpenEdge replaces the icon, shortcut menu options, or properties associated with the file extension with OpenEdge-specific information.

The shell integration DLL uses the DLC and, optionally, the PROMSGS environment variables to locate the PROMSGS file. The DLL searches these registry locations for the following variables:

- HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\10.1B\Startup
- HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\10.1B\Startup

The Installation Utility writes the proper values to the above registry locations. However, if after the installation you move OpenEdge to another location (or move or rename the PROMSGS file), you must edit the variables in the registry so that the shell integration DLL can find PROMSGS.

Icons

OpenEdge associates each of the OpenEdge-supported file types, except for compiled procedure code files (.r), with a unique icon that is displayed in Windows Explorer. You can execute the default action on a file by double-clicking on its icon. To perform other actions, you can right-click on the file and choose one of the options from the shortcut menu. To change the default setting, see the “[Shortcut menus](#)” section on page 3–22.

Shortcut menus

A shortcut menu allows you to perform an action on a file by eliminating several steps to accomplish the task. OpenEdge enhances this feature by adding context-specific options for each file type. For example, to edit the Sports database from Windows Explorer, right-click the sports.db icon and choose **Edit in Data Dictionary** (single-user) from the shortcut menu. If you do not use the shortcut menu, this same action requires several more steps.

To view the shortcut menu for a specific file, right-click the file. A shortcut menu appears with context-specific options.

To add to or change your default shortcut menu options, choose **View→Options→File Types** from Windows Explorer. In the **Registered file types** list, choose the OpenEdge file type you want to modify and click **Edit**.

The command line for each shortcut menu option includes a full PATH to the OpenEdge executable. If you move this executable to another location, you must modify the PATH.

Properties

By default, Microsoft provides general information about a file in its properties sheet. OpenEdge adds an extra page containing specific information for compiled procedure code (.r) and database (.db) file types. To view a file’s properties, right-click on the file and choose **Properties** from the shortcut menu.

UNIX-specific installation considerations

If any other required applications are necessary for your platform, install them before you install OpenEdge products. This section discusses:

- [Specific considerations.](#)
- [Upgrading an existing OpenEdge or Progress installation on UNIX platforms.](#)
- [Reviewing the UNIX system installation directory structure.](#)
- [Installing or viewing product documentation and samples.](#)

Specific considerations

Review the following specific considerations, depending on your operating system.

HP Tru64, AIX, Unixware, and Linux

If you are installing OpenEdge on a HP Tru64, or Linux machine, you must first install the required version of the JRE and/or JDK (if not already installed on your system). The directory where you install the JRE and/or JDK must be in your search PATH when you install and subsequently run OpenEdge. For more information on the JRE and JDK requirements, see [Chapter 2, “UNIX Systems Installation Requirements.”](#)

OpenEdge SQL

The installation program does not automatically install the JDK component when you install the OpenEdge Enterprise RDBMS, the OpenEdge Personal RDBMS, or the OpenEdge Workgroup RDBMS with SQL Client Access. (However, the program does automatically install the JRE.)

If you intend to develop Java stored procedures and Java triggers for your database, you must install an OpenEdge development product such as the 4GL Development System. For information on writing Java stored procedures and triggers, see [OpenEdge Data Management: SQL Development](#).

Upgrading an existing OpenEdge or Progress installation on UNIX platforms

If you have OpenEdge or Progress installed, you can upgrade to the most recent OpenEdge release.



To upgrade to OpenEdge:

1. Make sure that the `ULIMIT` is set to at least 8MB and at least 128 file descriptors. For specific instructions on setting the `ULIMIT` on your system, consult its man page by typing `man ulimit` at the command prompt.
2. Truncate the `before-image (.bi)` file of any existing database using the `PROUTIL TRUNCATE BI` utility. Back up your OpenEdge database using the `PROBKUP` utility. For more information on the `PROUTIL TRUNCATE BI` and `PROBKUP` utilities, see [OpenEdge Data Management: Database Administration](#).
3. Make copies of your `ubroker.properties` and `conmgr.properties` files to another directory. The new installation automatically upgrades the files in the `OpenEdge-install-dir/properties` directory. (However, note that the full `PATH` to `DLC` is not updated automatically; you must edit it manually.) After you have finished your new installation, replace the newly installed versions of these files with your copies. When you start the `AdminServer`, your older files will be updated to match the current standards for these files.
4. Make sure you are installing the software into a directory other than the directory from which you are running the Installation Utility.

Reviewing the UNIX system installation directory structure

The OpenEdge installation PATH contains configuration files and several subdirectories. The installation PATH directory contains the OpenEdge executables, several procedure (.p) files, and other related files and subdirectories. During installation, the \$DLC environment variable is automatically set to your OpenEdge installation PATH.

References to the UNIX or Linux installation directory in this guide

Throughout this guide, the installation PATH is referred to as either of the following:

- DLC** — The DLC variable on UNIX or Linux, \$DLC, is automatically set to your OpenEdge installation PATH. Historically, it has been a convenient way to refer to the location in which you have installed OpenEdge.

Note that the \$DLC variable is set in the various command scripts; the variable is not and should not be set at the system level. For information about the \$DLC environment variable, see [Chapter 8, “Working in the OpenEdge Environment on UNIX.”](#)

- OpenEdge-install-dir** — A more explicit phrase to refer to the directory location to which your OpenEdge installation PATH. The Windows environment variable DLC is also used to create this location; the use of the phrase *OpenEdge-install-dir* is intended to be more self-explanatory than is the reference \$DLC.

[Table 3–6](#) describes a directory tree of the OpenEdge subdirectories.

Table 3–6: OpenEdge-install-dir (\$DLC) directory structure (1 of 3)

Directory name	Description
bin	Contains the executable files for OpenEdge, such as PRODB. It also contains batch files and system executables.
certs	Contains the public keys of the Certificate Authorities (CAs) used by OpenEdge clients to perform server-side certificate validation when communicating with secure Web servers using HTTPS.
esbadapter	Contains the configuration and support code for the OpenEdge Adapter for Sonic ESB.
include	Contains the header files required for ESQ.

Table 3–6: OpenEdge-install-dir (\$DLC) directory structure*(2 of 3)*

Directory name	Description
install	Contains Java tailoring classes that only the Installation Utility uses. It also contains the uninstall script to remove an OpenEdge 10.1B installation.
java	Includes the Java files and executables necessary for running OpenEdge products.
javahelp	Contains .jar files for the OpenEdge Application Debugger.
jdk	Contains the Java Development Kit files and executables necessary for running OpenEdge products.
jms	Contains files to support client deployment of java messaging.
jre	Contains the Java Run-time Environment files and executables necessary for running OpenEdge products.
keys	contains encrypted RSA Private Key and Certificate file information.
lib	Contains shared objects necessary for running OpenEdge executables.
licenses	Contains license and copyright information related to HTTP Client, OpenSSL toolkit, Perl, and w3c IPR software notice.
odbc	Includes files to support ODBC.
oebuild	Includes files that the OpenEdge MAKE utility uses when creating custom executables.
ora	Contains files to support the DataServer for ORACLE.
perl	Contains files to support the use of the Perl scripting language.
proedit	Contains files to support the advanced editing features.
prohelp	Includes the online help and other necessary files for OpenEdge.
prolang	Contains the national language support directories.
properties	Contains property files that manage the configuration of OpenEdge services, such as WebSpeed, the NameServer, and the AppServer.
scripts	Can contain files related to the Failover Cluster component.

Table 3–6: OpenEdge-install-dir (\$DLC) directory structure *(3 of 3)*

Directory name	Description
servlets	Identifies the default location of the AppServer Internet Adapter (AIA) and Web Services Adapter (WSA) servlet containers. These containers include Web definitions. ¹
sonic	Contains files that support the Sonic client and container.
sports	Includes the schema triggers and supplier information for each sample database.
sports2000trgs	Includes the schema triggers for the Sports2000 database.
src	Contains source files for OpenEdge ADE tools, such as the Data Dictionary, Procedure Editor, and Sample Applications.
templates	Can contain files related to the Failover Cluster component.
toolkit	Includes files that help in deploying and encrypting your applications.
tty	Includes mostly object (.r) files and r-code procedure libraries (.pl) for character-mode OpenEdge.
ubqmanager	Includes files used by the AppServer exclusively. Do not modify these files.
uninstall	Contains the uninstall script which allows users to uninstall third-party software embedded in the OpenEdge installation, remove release-specific entries from other files, and remove the install directory into which the OpenEdge 10.1B installation has been installed.
webspeed	Includes static files for the WebSpeed Workshop, WebTools, and HTML help that reside on the Web server.

1. Refer to your OpenEdge product documentation for details about configuring AIA and WSA.

Installing or viewing product documentation and samples

The Documentation and Samples CD is shipped with your Release 10.1B media box and it contains the following information:

- OpenEdge Release 10.1B PDF-formatted documentation and OpenEdge sample files.
- OpenEdge Management Release 3.1B PDF-formatted documentation.
- OpenEdge Replication Release 10.1B documentation.

This section presents the procedures to install the documentation and sample information in Windows and on UNIX platforms.

Installing documentation and samples in Windows platforms

Obtain the Documentation and Samples CD from the Release 10.1B media box to proceed. The InstallShield Wizard prompts you through the documentation and samples installation process.



To install documentation and samples from the Documentation and Samples CD:

1. Insert the Documentation and Samples CD into your CD drive shown in the following table. Proceed:

If you . . .	Then . . .
Have autorun enabled.	The installation program starts automatically.
Do not have autorun enabled.	From Windows Explorer, browse to the root directory on the CD and run <code>setup.exe</code> .

The **OpenEdge Documentation and Samples - InstallShield Wizard for OpenEdge 10.1B** dialog box appears. Click **Next**.

Note: If you want to install the sample files into your Progress OpenEdge installation directory, Progress OpenEdge software must already be installed. Otherwise, the samples are installed only to the Documentation and Samples CD installation location.

2. The **Select Components** dialog box appears, displaying a checkmark next to each installable component. The following list identifies the components that appear on the dialog box:

- **OpenEdge 10.1B Documentation and Samples**
- **OpenEdge Management 3.1B Documentation**
- **OpenEdge Replication 10.1B Documentation**

Proceed as shown in the following table:

If you . . .	Then . . .
Plan to install all components.	Leave the check marks and go to Step 3 in this procedure.
Do not want to install all components.	Click the checkmark next to the components you do not want to install. Proceed to Step 3.

3. Choose the installation directory, or select the default location:
C:\Progress\OpenEdge_101B_Doc.
4. Click **Disk Space** to review **Disk Space Required** and **Disk Space Available** details for the drive you select.
5. Click **Next**. The **Start Copying Files** dialog box appears.
6. On the **Start Copying Files** dialog box, click **Next**. The components which have checkmarks are installed.
7. The **Install Wizard Complete** dialog box appears after all files have been installed. Click **Finish** to dismiss the Install Wizard.

8. Once you have installed your OpenEdge 10.1B products, select **Start→Programs→OpenEdge Documentation** to access a component you installed.

Viewing documentation directly from the Documentation and Samples CD

You can also view the PDF documentation files directly from the CD.



To view documentation directly from the CD:

1. Install the Documentation and Samples CD in the CD drive.
2. Browse to the following directories to access the **Welcome** page associated with a documentation set:
 - OpenEdge_Doc/openedge/start.pdf
 - OpenEdge_Doc/fm/start.pdf
 - OpenEdge_Doc/fr/start.pdf

Installing documentation and samples on UNIX platforms

Obtain the Documentation and Samples CD from the Release 10.1B media box to proceed. The InstallShield Wizard prompts you through the documentation and samples installation process.



To install documentation and samples from the Documentation and Samples CD:

1. Mount the Documentation and Samples CD into your CD drive.
2. Create a folder on the machine where you want to install the documentation and samples. For example, you can create a folder named OpenEdge_101B_Doc.

3. Proceed as follows:

If you plan to install . . .	Then . . .
All documentation and sample data available from the CD.	Copy the entire OpenEdge_Doc folder to the file location you created in Step 2 of this procedure.
Some of the documentation and sample data from the CD.	<p>Copy only the directories that you want to install, as described in the following list:</p> <ul style="list-style-type: none">• fm — Contains the OpenEdge Management documentation.• fr — Contains the OpenEdge Replication documentation• openedge — Contains the OpenEdge documentation• src, tutorial, and webinstall — Contains the sample code and documentation example files. (You might also want to copy these same directories to your OpenEdge installation directory.)

4. After you have installed your documentation and/or samples, you can open the **Documentation Welcome** pages located in the following directories:

- OpenEdge_Doc/openedge/start.pdf
- OpenEdge_Doc/fr/start.pdf
- OpenEdge_Doc/fm/start.pdf

WebSpeed configuration choices

WebSpeed is an OpenEdge component to develop and deploy Web browser-based online transaction processing (OLTP) for business applications. WebSpeed products require a Web server product for which you must configure a Web server directory on a machine where you want these products to reside. If you are upgrading an existing WebSpeed installation, shut down your Web server and reboot your machine. For information about shutting down a Web server and examples of various WebSpeed configurations, see *OpenEdge Getting Started: WebSpeed Essentials*.

Developing Web applications with WebSpeed

These OpenEdge 10.1B products support developing Web applications with WebSpeed:

- OpenEdge Studio (includes WebSpeed Workshop and Progress Dynamics®).
- WebSpeed Workshop.
- OpenEdge Architect (includes the WebSpeed Workshop).
- OpenEdge Development Server (includes WebSpeed Transaction Server).

Deploying Web applications with WebSpeed

These OpenEdge 10.1B products support deploying Web applications with WebSpeed:

- OpenEdge Application Server Basic (includes WebSpeed TransactionServer).
- OpenEdge Application Server Enterprise (includes WebSpeed Transaction Server).
- WebSpeed Messenger (which is a component of the Application Server products).

To choose the proper configuration to install, you need to consider your network resources and whether you want to create a development, deployment, or combined development and deployment configuration. You can distribute the components required to run WebSpeed in many ways. *OpenEdge Getting Started: WebSpeed Essentials* also describes the WebSpeed components and how they work together.

Performing an OpenEdge Installation in Windows

This chapter contains the instructions for installing OpenEdge in Windows platforms, as described in the following sections:

- [Installation overview](#)
- [Running the Progress Dynamics Configuration Utility](#)
- [Additional product installation activities](#)
- [OpenEdge Silent installation overview](#)
- [Performing postinstallation tasks](#)
- [Uninstalling OpenEdge in Windows](#)
- [Sharing an OpenEdge installation on a network overview](#)
- [Uninstalling the Shared Network Installation Utility](#)
- [Running the Silent installation option for the Shared Network Installation Utility](#)

Installation overview

After you have addressed all the topics presented in the “[Tasks overview](#)” section on page 3–2, you are prepared to perform the OpenEdge installation in a Windows platform.

Loading the installation media

You must have Administrator privileges on the machine where you are installing OpenEdge. For more information, see your Windows documentation.



To initiate the Installation Utility to install OpenEdge products:

1. Obtain a copy of the completed *Preinstallation Checklist for Windows*. You might also want to have handy the other installation-related documents highlighted in [Table 3–1](#) of the “[Gathering information to plan your installation](#)” section on page 3–3.

Note: When you install a client networking license, the ADM2 directory is not installed in the `OpenEdge-install-dir\GUI` directory. This r-code is considered part of your application and should be deployed as a module of your application.

2. Close all other applications before beginning the installation process.

Other applications or tasks might interfere with the installation or use files that OpenEdge needs to complete the installation. Shut down any processes where the executable itself, or a file used by the executable, is located in the directory where you intend to install OpenEdge.

3. Install the installation program from the installation medium you plan to use, as described in the following table:

For this installation medium . . .	Do the following . . .
CD	Insert it into the CD drive. If the CD does not run automatically, double-click <code>setup.exe</code> in the root directory of the CD.
DVD	Insert it in the DVD player and navigate the directory structure, locating the specific platform directory to which you intend to install. ¹ Double click the <code>setup.exe</code> to start the Installation Utility.
Electronic Software Distribution (ESD) download	Navigate to the software image you intend to download from the Progress Software Download Center. ²

1. The installation DVD contains subdirectories for all Windows and UNIX platforms that OpenEdge 10.1B supports. However, only the specific platform and type to which your license pertains is installable.
2. The Progress Software Download Center is available at <http://www.progress.com/esd>. Access to Progress software products and updates at this Web site requires a valid account.

Performing the installation

Once you have loaded the installation program from your installation medium, you are ready to perform the online tasks required to install OpenEdge.

Refer to [Table 3–1](#) for the documents you should reference during the installation to help you perform the online OpenEdge installation.

Also, refer to the online installation help system which contains a help topic for each installation dialog box. To access the online help while you are running the Installation Utility:

- Choose **Help** on an installation dialog box. The help topic associated with the dialog box appears and describes the step-by-step procedure to perform to complete the dialog box.
- Choose help topics that display in the help system's Table of Contents. Note that the help viewer in which you can read an individual help topic also displays the help system's Table of Contents in the left pane. Use the Table of Contents to navigate through all the online installation-related help topics. To display the Table of Contents, click **Show** on the Navigator bar.

Finishing the installation

If you saved information in a `progress.ini` file in a previous version of OpenEdge that you want to continue using, you can add that information to the new `progress.ini` file. Perform the following procedure whether you installed using the online, interactive method or the silent method.



To add information about your `progress.ini` file:

1. Copy the information you want to save from your previous `progress.ini` file to the OpenEdge `progress.ini` file.
2. If you are copying information from a Version 9 installation, rename any `PROUIB` section to `PROAB`. The `PROUIB` section of the Version 9 `progress.ini` file is referred to as `PROAB` in OpenEdge.
3. Run `ini2reg` to update the information in the registry with the information you added from your previous `progress.ini` file.
4. Restart your system.

Using property information from a previous installation

If you want to continue using the property information (such as `ubroker.properties`, `conmgr.properties`, or `proxygen.preferences`) that OpenEdge automatically saved prior to the current installation, copy the saved property files from `%TEMP%` to `OpenEdge-install-dir\properties`. For information about the automatic save of the property information before the installation process occurs, see the [“OpenEdge automatic save of Properties files”](#) section on page 3–17. For information on merging property files, see the [“Mergeprop utility overview”](#) section on page 10–26.

Also, you will need to perform any other postinstallation tasks as discussed in the [“Postinstallation considerations”](#) section on page 4–5 and the [“Performing postinstallation tasks”](#) section on page 4–41.

Postinstallation considerations

Note these points after you have performed the Installation Utility:

- If you installed a product that included Progress Dynamics, the Progress Dynamics Configuration Utility (DCU) must be run as a postinstallation task. See the “[Running the Progress Dynamics Configuration Utility](#)” section on page 4–5.
- Before you can start OpenEdge, the AdminServer must be running. For information about the AdminServer, see the “[Getting started with the AdminServer](#)” section on page 7–11.
- Detailed product information about the OpenEdge products you installed is available in the Release 10.1B product documentation set. Access the PDF-formatted documentation set from either the OpenEdge Documentation and Sample CD or the Documentation link on the Progress Software Corporation Web site at <http://www.psdn.com/library/kbcategory.jspa?categoryID=129>

Running the Progress Dynamics Configuration Utility

To complete the Progress Dynamics ® installation, OpenEdge provides the Progress Dynamics Configuration Utility (DCU). The DCU is an Advanced Business Language program.

The DCU completes the installation by building a new icfdb Repository database or by upgrading an existing one from a previous release. This section includes the following:

- [Before you begin.](#)
- [Completing the DCU wizard.](#)
- [Editing Progress Dynamics files.](#)

Before you begin

Note these points about the Progress Dynamics Configuration Utility before you perform the procedures presented in this section:

- The DCU completes the installation by building a new `icfdb` Repository database or by upgrading an existing one from a previous release. (Consult the *Release Notes* for the most specific details about upgrading to the latest Progress Dynamics release; this step is most important to users who are upgrading from earlier versions of the DCU.)
- The DCU does not require ABL to run. You can use the DCU to deploy Progress Dynamics to client sites that do not have the compiler installed.
- The Progress Dynamics Configuration Utility launches directly after the OpenEdge installation concludes provided the following conditions are met:
 - You are installing Progress Dynamics as a component of either OpenEdge Studio or OpenEdge Architect.

Note: To install Progress Dynamics as a component of OpenEdge Architect, you must select the component on the **Configuring / Installing Components** dialog box. The Progress Dynamics component supports the AppBuilder functionality within OpenEdge Architect. Therefore, you must select the **Progress Dynamics** option on this dialog box to install it to enable Progress Dynamics in OpenEdge Architect. For more information, see the [Appendix A, “Preinstallation Checklist for Windows.”](#)

- During the OpenEdge installation, you select the **Install/upgrade Dynamics repository** option on the **Progress Dynamics Options** dialog box.

After you choose **Finish** in the InstallShield’s **Complete Setup Done** dialog box, an OpenEdge session starts up. Then, DCU wizard starts by displaying the **Progress Dynamics Configuration Utility - Welcome** dialog box.

The DCU performs its work in a progressive and re-entrant fashion. If for any reason the DCU does not complete its work, or if you quit the utility before it has a chance to finish, you can rerun it (from the command line or a shortcut) to complete its work. For more information about the DCU, see *OpenEdge Development: Progress Dynamics Administration*.

Note: The DCU does not remove any part of a Progress Dynamics installation.

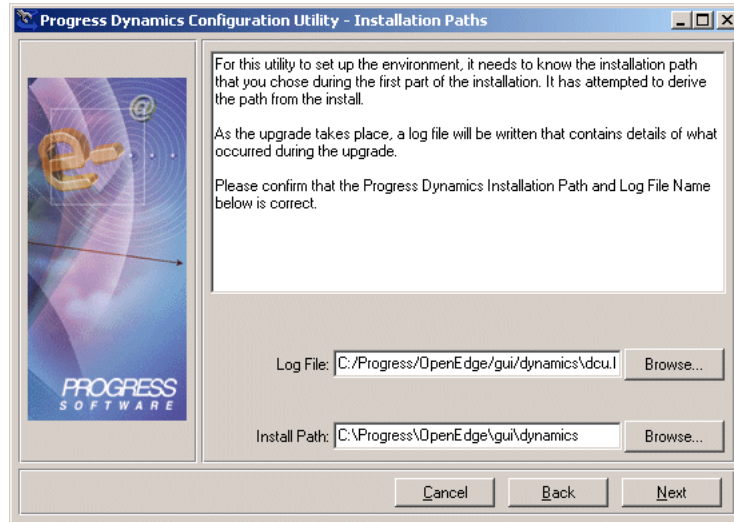
Completing the DCU wizard

This section presents the procedures to complete the Progress Dynamics installation using the Progress Dynamics Configuration Utility (DCU).



To complete the DCU wizard:

1. Choose **Next**. The **Installation Paths** dialog box appears:

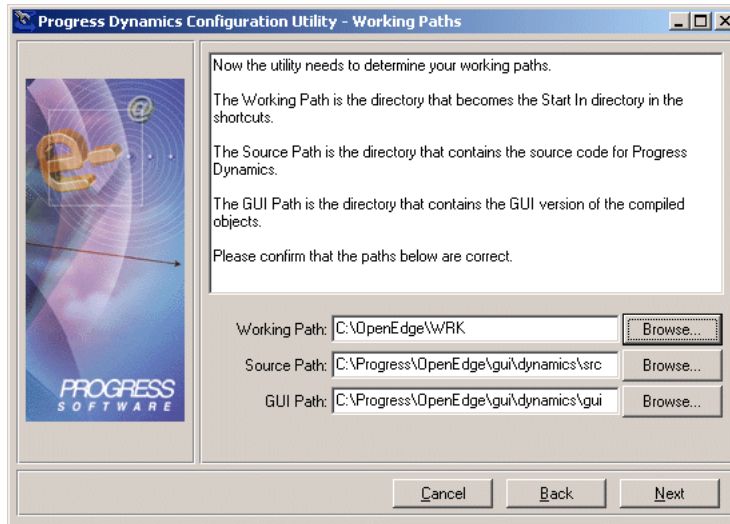


2. Enter the path of the directory where Progress Dynamics is installed.

The **Install Path** field contains a default entry based on the information you provided when you installed Progress Dynamics.

3. Enter the path for the DCU log file. The **Log File** field contains the default pathname.

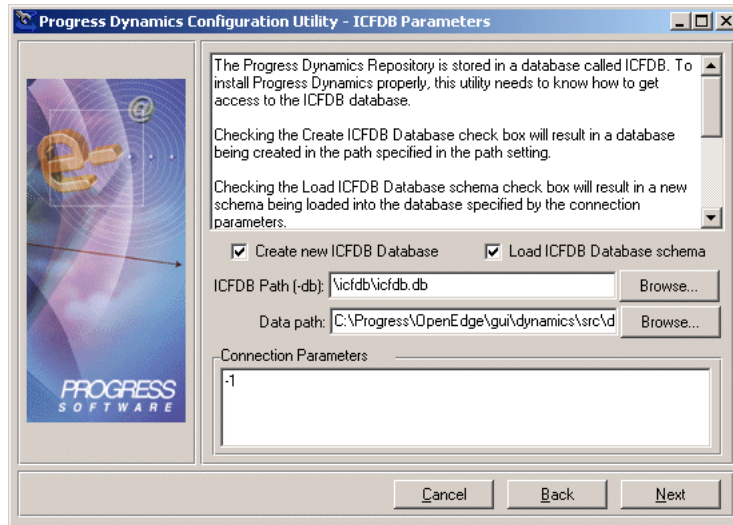
4. Choose **Next**. The **Working Paths** dialog box appears:



5. Enter the path information in the **Working Path**, **Source Path**, and **GUI Path** fields. The fields contain default entries based on the information you provided when you installed OpenEdge.

Note: The **Working Path** must **not** be under your OpenEdge installation directory. Otherwise, you can lose all of your work during future installs and upgrades of these products. Enter the path of your `icfdb` database and the database connection parameters.

After you enter the appropriate path information, choose **Next**. The **ICFDB Parameters** dialog box appears:



For more information on connection parameters, see [OpenEdge Deployment: Startup Command and Parameter Reference](#).

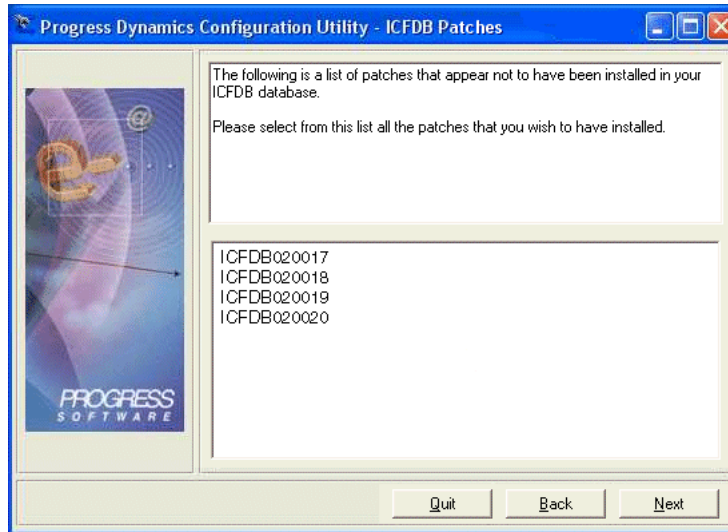
If you try to use the **Create New ICFDB Database** option to create a database on a remote machine, you will get an error message. To load the `icfdb` schema to a remote database, you first need to create the database on the remote machine by starting a client and running the PRODB utility. For more information on the PRODB utility, see [OpenEdge Data Management: Database Administration](#).

Caution: If you are upgrading, you should remove the check from both the **Create new ICFDB Database** and the **Load ICFDB Database schema** options.

Also note the following if you are upgrading:

- If you check only the **Create new ICFDB Database** option, the DCU automatically loads the database schema as well. The DCU also deletes any database that already exists in the specified `icfdb` path and creates a new one.
- If you check only the **Load ICFDB Database schema** option, the DCU treats any existing `icfdb` as if it was empty and loads the schema.

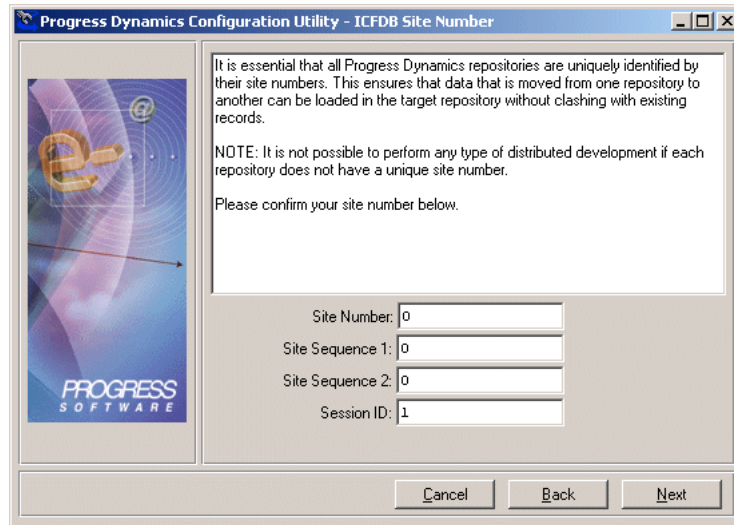
After you enter the appropriate information, choose **Next**. If you are upgrading, the **ICFDB Patches** dialog box appears:



If this is a new installation, the **ICFDB Patches** dialog box does not appear and you can proceed to [Step 8](#).

6. Review the patch information. The DCU lists which patches are needed to upgrade your database to the latest level. The list will vary depending on what patch level was added to the previous version of your `icfdb` database. The DCU applies the patches you select patches in the correct order.

7. After you review the patch information, choose **Next**. The **ICFDB Site Number** dialog box appears:



Progress Dynamics Configuration Utility - ICFDB Site Number

It is essential that all Progress Dynamics repositories are uniquely identified by their site numbers. This ensures that data that is moved from one repository to another can be loaded in the target repository without clashing with existing records.

NOTE: It is not possible to perform any type of distributed development if each repository does not have a unique site number.

Please confirm your site number below.

Site Number: 0

Site Sequence 1: 0

Site Sequence 2: 0

Session ID: 1

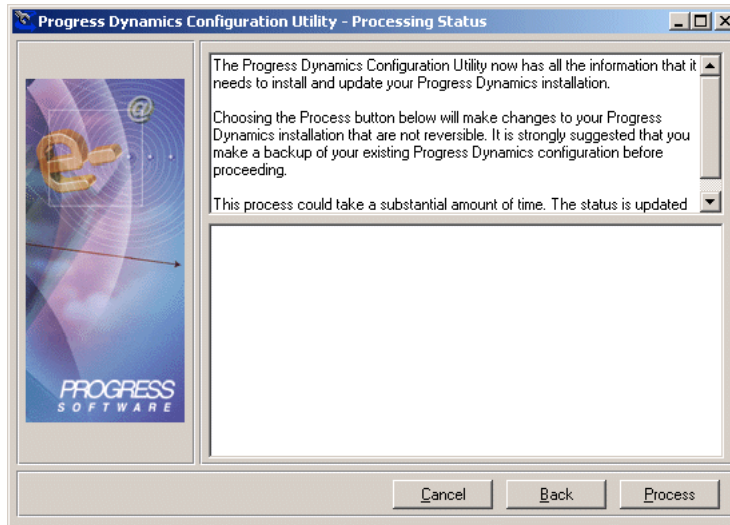
Cancel Back Next

8. Enter appropriate values in the **Site Number**, **Site Sequence 1**, **Site Sequence 2**, and **Session ID** fields. For more information, see [OpenEdge Development: Progress Dynamics Administration](#).

If you are upgrading, the DCU transfers sequence values from the previous version of the Repository. Or you can obtain these values from the **Set Site Number** dialog box. You can access the **Set Site Number** dialog box from the Progress Dynamics AppBuilder **Build** menu.

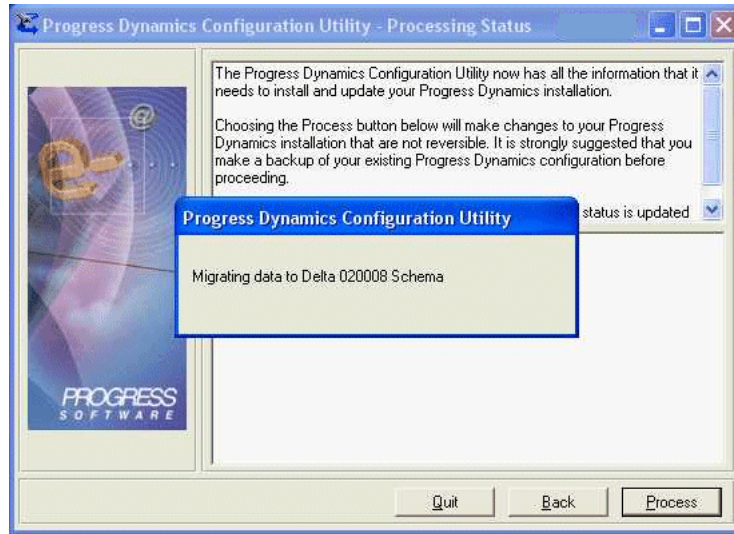
Note: Verify that the **Site Sequence 1** value is increased when you are upgrading. If the **Site Sequence 1** value in this installation is less than the value in the previous release, you will get error messages stating that an Object ID is already used.

9. After you enter the appropriate values, choose **Next**. The **Processing Status** dialog box appears:



Caution: If you are upgrading, this is a good time to make a backup of your existing Progress Dynamics configuration because you might want to refer back to your prior configuration. The DCU processing that occurs next is irreversible. You will lose configuration information if you do not make a backup before running the DCU.

10. Choose **Process** to start the DCU. The **Processing Status** dialog box appears:



The appearance of this dialog box indicates which upgrade program is currently running.

Click **Finish** when the DCU completes processing.

Editing Progress Dynamics files

Typical of installations that use Progress Dynamics, you will want to connect to the Progress Dynamics Repository database over a network. This type of configuration requires that you edit your services file to identify the service name and a port number for the icfdb Repository database.

The first time you install Progress Dynamics, assign a port number to the Repository Database server in the Windows services file. Subsequent OpenEdge uninstalls do not remove this entry. Therefore, you should only need to perform this task once.



To edit your services file:

1. In a text editor, open your services file. By default, the services file is in the C:\WINNT\System32\drivers\etc\directory. (In Windows XP Professional, the directory is C:\WINDOWS\System32\drivers\etc\.)
2. Assign port numbers, using the following format:

```
service_name  port_number/tcp
```

The *service_name* is the name you specify with the -S parameter when you start the database. The *port_number* is a unique four-digit number (one that is not already assigned to another service in the file). For example:

```
icfdb  8000/tcp
```

3. Add an additional line for each of your application databases, using a unique service name and a unique port number for each one.
4. Save and close the services file.

Caution: You cannot start two Repositories if both have the same service name (icfdb for example). If you need to run more than one Repository database, each version must have a different service name and a different port number.

Editing installed files

Table 4–1 lists the files installed by the Progress Dynamics installer program that you can edit to modify configuration settings, add databases, or change paths.

Table 4–1: Progress Dynamics files that you can edit

Filename	Description	Location
icfconfig.xml	An XML file containing the instructions used by the Progress Dynamics Configuration File Manager to start up a specified Progress Dynamics session. For more information, see OpenEdge Development: Progress Dynamics Administration .	<i>OpenEdge-install-dir\gui\dynamics</i> Default: C:\Progress\OpenEdge\gui\dynamics
startdb.bat	Starts database servers for the Progress Dynamics Repository by invoking proserve.	<i>OpenEdge-install-dir\bin</i> Default: C:\Progress\OpenEdge\bin
stopdb.bat	Stops database servers for the Progress Dynamics Repository by invoking proshut.	
icf.ini	The Progress Dynamics initialization file (the Progress Dynamics version of the progress.ini file).	
icf.pf	The Progress Dynamics AppBuilder startup parameter file.	<i>OpenEdge-install-dir</i> Default: C:\Progress\OpenEdge

If you are upgrading from an earlier version of Progress Dynamics, there are several additional tasks that you need to perform. To ensure that your existing applications run under the newer release, review and complete the tasks described in the following sections:

- [Editing the Progress Dynamics XML configuration file.](#)
- [Starting a development session.](#)
- [Stopping and restarting Progress Dynamics.](#)
- [Updating session types.](#)
- [Running the Entity Import tool.](#)
- [Recompiling application code.](#)
- [Setting up for Web development.](#)

Editing the Progress Dynamics XML configuration file

The DCU upgrade process is not complete at this point. The DCU must run again to apply other upgrade procedures and to update data sets for the newly created `icfdb`. The DCU automatically runs again when you start an administrative session. But before you can log in and start an administrative session, you must have an XML configuration file that is compatible with the newer version of Progress Dynamics.

If you did not create a new `icfconfig.xml` file (or edit the default) for your application, you can skip this section. You can simply use the standard `OpenEdge-install-dir\gui\dynamics\icfconfig.xml` file that ships with Progress Dynamics to access the upgraded Repository.

However, if you modified or changed the name of the default XML configuration file (for example, you might have a customized XML configuration file in your current directory), you must edit the XML configuration file to make it compatible with your upgraded application. For example, you must add service entries for the new managers, and change the connection parameters for your `icfdb` database.

You should only edit the `icfconfig.xml` file for the session type that you use as your administrative session type. This allows you to connect to your Repository with administration privileges. Then you can use the Dynamics Administration tool's **Session** menu options to modify your other session types and regenerate the `icfconfig.xml` file.

**To edit your icfconfig.xml file:**

1. Create a backup copy of icfconfig.xml before you edit it.

Caution: If you make a mistake in the following steps, you might render your icfconfig.xml file unreadable and your Progress Dynamics session might not start. Therefore, creating a backup copy that you can revert to is extremely important.

2. Open your icfconfig.xml file in a simple text editor, such as Notepad.
3. Search for the string: SessionType=ICFDev where ICFDev is the name of the session type that you use for administration tasks. This string should occur inside a session node as follows:

```
<session SessionType=ICFDev>
```

Immediately following the <session> node is a <properties> node.

4. Scan down the file until you pass the end of the properties node, which is denoted by the end properties tag (</properties>).

Immediately following the <properties> node is a <services> node. The <services> node contains the list of services that should be connected when the session starts. Among them is a <service> node for each of the databases and AppServers that are connected to this session. Each </service> node is contained within a start <service> and end </service> tag.

5. Search for a <cServiceName> tag with the value rvdb.

If it exists, remove the entire <service> node for the RVDB service type from the file.

The rvdb database was used in Progress Dynamics Version 1.1A. It became obsolete in Version 2.0A. There should be no references to it in any of your configuration files.

6. Scan down until you find the first end managers (</managers>) tag.

7. Insert the following XML statements immediately before the line noted in Step 6 of this procedure:

```
<manager>
<cManagerName>RIManager</cManagerName>
<cFileName>ry/app/ryrisrvrp.p</cFileName>
<cHandleName>RI</cHandleName>
<cSuperOf/>
</manager>
<manager>
<cManagerName>CustomizationManager</cManagerName>
<cFileName>ry/app/rycusrvrp.p</cFileName>
<cHandleName>NON</cHandleName>
<cSuperOf/>
</manager>
<manager>
<cManagerName>RepositoryDesignManager</cManagerName>
<cFileName>ry/app/rydessrvrp.p</cFileName>
<cHandleName>NON</cHandleName>
<cSuperOf/>
</manager>
```

8. Scan down until you find a </service> node that contains a <cServiceName> tag with the value icfdb.
9. Change the database connection parameters from the values for the earlier version to the appropriate values for the newer version.

The arguments for the -db and the -S parameters should be the icfdb that you upgraded through the DCU.

The bold text in the following sample shows the changes to the icfdb services entry:

```
<service>
  <cServiceType>Database</cServiceType>
  <cServiceName>ICFDB</cServiceName>
  <cPhysicalService>ICFDBn</cPhysicalService>
  <cConnectParams>-db icfdbv21A -N TCP -H localhost
  -S icfdbv21A</cConnectParams>
  <lDefaultService></lDefaultService>
  <lCanRunLocal></lCanRunLocal>
  <iStartOrder></iStartOrder>
</service>
```

Since the upgrade simultaneously opens a large number of records, it is possible that you might get an error stating that the record lock table is too small. In that case, you must set the Lock Table Entries parameter (-L) to a very large value. (A value of 500,000 should be adequate.) See [OpenEdge Deployment: Startup Command and Parameter Reference](#) for more information.

10. Save the edited `icfconfig.xml` file.
11. Place the `icfconfig.xml` file in a directory that is included in your `PROPATH`.

Starting a development session

The DCU must run again to apply other upgrade procedures and to update data sets for the newly created `icfdb`. The DCU automatically runs again when you start a session that has administrative privileges. By default, you can run the development session (ICFdev) with administrative privileges. When you ran the DCU the first time, it wrote information to the Repository, which it now uses to complete the upgrade.

To update data sets, Progress Dynamics applies Application Dynamic Object (ADO) files to the Repository. ADO files are XML documents that have a `.ado` filename extension.

After you complete the tasks described in the [“Editing the Progress Dynamics XML configuration file”](#) section on page 4–16, you should be able to start a session that connects to the `icfdb` that the DCU upgraded.



To start an administrative session:

1. Start the DB servers for the newer version of Progress Dynamics, if they are not already running.
2. Start an administrative session, logging in as **admin**.

Note: No password is required.

If you start a Progress Dynamics Development session from a desktop shortcut, check the properties to make sure that the value of ICFSESSTYPE is set to your administrative session type (usually ICFDev). Also, verify that the `-ini` and `-pf` parameters point to the Release 10.01A initialization and startup parameter files. For example:

```
C:\Progresss\OpenEdge\bin\prowin32.exe -p icfstart.p  
-pf "C:\Progress\OpenEdge\icf.pf"  
-ini "C:\Progress\OpenEdge\bin\icf.ini"  
-icfparam ICFSESSTYPE=ICFDev
```

After you log in, the DCU starts, runs upgrade programs, and applies ADOs. The DCU displays a status window that indicates its **progress**.

This phase of the upgrade involves running a large number of procedures, and it can be very time consuming. The actual duration depends on the size and complexity of your application.

Stopping and restarting Progress Dynamics

After the DCU loads all of the ADOs, you will be in a Dynamics AppBuilder session. Before you can continue, you must apply the changes you made up to this point by stopping and restarting Progress Dynamics. Log out of the current session. When you log in again, you must start an administrative session, as described in the [“Starting a development session”](#) section on page 4–19.

Updating session types

In the “[Editing the Progress Dynamics XML configuration file](#)” section on page 4–16, you updated the session that you use to administer your Repository. You must now update any other session types that apply to your application.

From the **Dynamics Administration Tool Session** menu, access the **Session Type Control** and modify your existing session types. After modifying your session types, regenerate the XML configuration file. (You can also edit the configuration file manually, but manual editing is more error prone.) See [OpenEdge Development: Progress Dynamics Administration](#) for more information about defining, modifying, and managing sessions.

Note: If you did **not** create a new `icfconfig.xml` file (or edit the default) for your application, you can skip this section. You can simply use the standard `icfconfig.xml` file that ships with Progress Dynamics to access the upgraded Repository.

Customizing Progress Dynamics session types

When you customize session types, you must add the appropriate managers. [Table 4–2](#) identifies the managers required for certain functionality.

Table 4–2: Managers for customized session types

(1 of 2)

Add the . . .	For session types that . . .
Referential Integrity Manager (RIManager) (ry/app/ryrisrvrp.p),	Make database connections.
Customization Manager: the server side manager (ry/app/rycussrvrp.p); the client-side manager (ry/app/rycusc1ntp.p ,	Make use of the customization facilities.
	All development session types.
	All server-side (AppServer and WebSpeed) session types.
RepositoryDesignManager (ry/app/rydessrvrp.p) ¹ ,	Design objects in the appbuilder.

Table 4–2: Managers for customized session types (2 of 2)

Add the . . .	For session types that . . .
RequestManager (ry/app/ryreqsrvr.p),	Handle requests coming in from a Web browser.
UserInterfaceManager (ry/app/ryuimsrvr.p),	Handle the interactions with the user interface.

1. The Repository Design Manager is only needed for development session types. It is not needed and should not be included in deployment sessions. It will only add unnecessary overhead when the application runs.

If you plan to deploy your application as a browser-based application on the Web, you must create an ICFWS session type. (See *OpenEdge Development: Progress Dynamics Administration* and *OpenEdge Development: Progress Dynamics Web Development Guide* for more information.)

The following example shows entries for all the new managers as they appear in the XML configuration file:

```
<manager>
  <cManagerName>RIManager</cManagerName>
  <cFileName>ry/app/ryrisrvrp.p</cFileName>
  <cHandleName>RI</cHandleName>
  <cSuperOf/>
</manager>
<manager>
  <cManagerName>CustomizationManager</cManagerName>
  <cFileName>ry/app/rycusrvrp.p</cFileName>
  <cHandleName>NON</cHandleName>
  <cSuperOf/>
</manager>
<manager>
  <cManagerName>RepositoryDesignManager</cManagerName>
  <cFileName>ry/app/rydessrvrp.p</cFileName>
  <cHandleName>NON</cHandleName>
  <cSuperOf/>
</manager>
<manager>
  <cManagerName>RequestManager</cManagerName>
  <cFileName>ry/app/ryreqsrvrp.p</cFileName>
  <cHandleName>NON</cHandleName>
  <cSuperOf/>
</manager>
<manager>
  <cManagerName>UserInterfaceManager</cManagerName>
  <cFileName>ry/app/ryuimsrvrp.p</cFileName>
  <cHandleName>NON</cHandleName>
  <cSuperOf/>
</manager>
```

Running the Entity Import tool

When the DCU runs after the initial installation, it displays an important note that you must run the Entity Import Tool with **Override all attributes from schema** selected. Your upgraded application database might not run correctly if you do not run the Entity Import Tool.

Start the Entity Import Tool from the main menu of the Administration Tool. Select **System→Entity Import**.

In later versions of Progress Dynamics, DataFields are used extensively throughout the tools and at run time in Progress SmartDataObjects for attributes such as formats, data types, and labels. *DataFields* are a level of abstraction from the physical data storage. When you upgrade, you must run the Import Entity Tool to ensure that Progress SmartObjects are created for every entity, with DataField instances representing the fields that belong to the entity.

In addition, due to the increased and fundamental use of DataFields, it is imperative that you keep them up-to-date. Whenever you make schema changes to your application database, you must use the Entity Import tool to update your application's entities and DataFields. You must run the Entity Import tool against the central database for your organization. (If individual developers run the Entity Import on their own "satellite" databases, they might generate different Object IDs for the same DataField.)

By default, the Entity Import process does not overwrite any local changes you have made to attributes, such as labels, but if the value of your label matches the value of the old schema label, the Entity Import process updates the DataFields appropriately. The Entity import tool includes an **Override all attributes from schema** toggle box. If you select this option, the Entity Import process overwrites the local changes you made to the values of the entity attributes with the schema values.

If you do not update the DataFields, then at run time the SDOs use the old values in the DataFields, thereby regressing your schema changes at run time.

Note: The generateDataFields API in the Repository Design Manager supports the overriding of local attribute values with database metaschema values for certain attributes, such as Format, Label, and Help.

Recompiling application code

Any static application code (in product modules, for example) must be recompiled in OpenEdge. In particular, Progress Dynamics Version 1.1A code will not run in an OpenEdge environment. However, you should recompile the application code from any previous release.

Setting up for Web development

If you plan to use Progress Dynamics to create applications for the Web, you should:

- Test the Broker/Agent setup.
- Test the Managers.
- Test the StartUp page.

For detailed information about setting up and creating Web applications with Progress Dynamics, see *[OpenEdge Development: Progress Dynamics Web Development Guide](#)*.

Additional product installation activities

This section highlights the following additional product-related activities you might also want to perform:

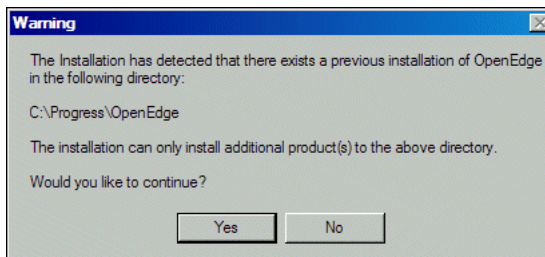
- [Installing additional products.](#)
- [Installing additional components to previously installed products.](#)
- [Viewing registry information.](#)
- [Downloading executables for heterogeneous environments.](#)

Installing additional products

You can add other OpenEdge products to your current installation by following the steps outlined in the “[Installation overview](#)” section on page 4–2.

Note: The AdminServer must be shut down before you can successfully add additional products to a current installation.

When the installation process detects the existing version of OpenEdge, a **Warning** dialog box appears, notifying you of the existing version’s location, as shown:



Note: When you add products to an existing installation, you can use the installation utility in batch mode regardless of the type of installation (complete or custom) that you are performing.



To continue with the installation:

1. Choose **Yes** to continue with the installation. The **Welcome** dialog box appears.
2. Choose **Next** to continue. The **Serial Number and Control Codes** dialog box appears.
3. Enter the serial number and control numbers and choose **Accept** for each product you want to add to your current installation.
4. Choose **Next** to continue. The **Progress License Agreement** dialog box appears.
5. Review the information and choose **Yes**. The **Choose Destination And Working Path Directories** dialog box appears. The install program deactivates (grays out) the **Browse** associated with the **Destination Directory** field and adds your OpenEdge products to directories automatically.
6. Accept the default directories and continue with the installation.

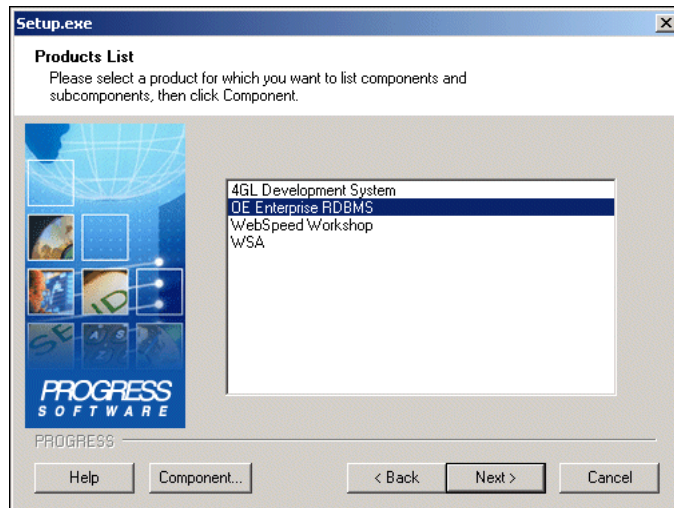
Installing additional components to previously installed products

You can add components and subcomponents to existing OpenEdge and later installations without entering any data other than the required components or subcomponents. In earlier Progress versions it is necessary to reinstall Progress and execute the “Custom Install” setup type; these steps are removed with this Add feature.



To add components or subcomponents to a previously installed product:

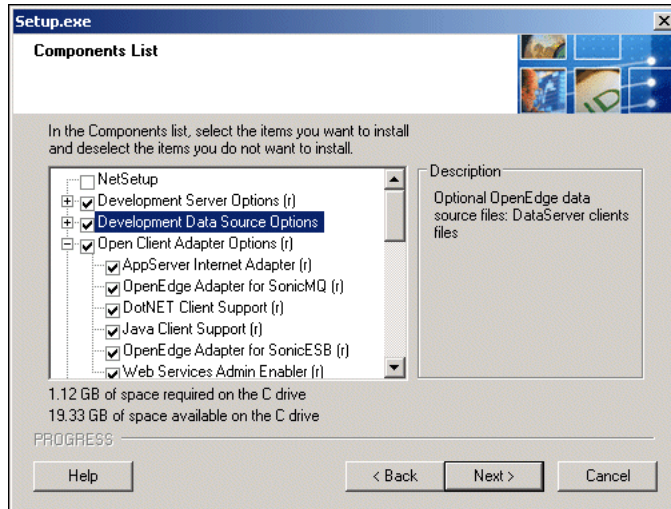
1. Choose **Start→All Programs→OpenEdge→Add Components**. The **Products List** dialog box appears:



All previously installed products appear on the **Products List**.

2. Select the product to which you want to add components or subcomponents from the **Products List**.

3. Choose **Component** to add components or subcomponents to the already installed product. The **Components List** dialog box appears:



Only components and subcomponents that you have not previously installed appear in the **Components List** dialog box.

4. Select the components and subcomponents you want to add.
5. Repeat [Step 2](#) through [Step 4](#), as needed.
6. Choose **Next**.
7. Choose **Finish** to update the existing install.

Note: If a system file in a newly added component or subcomponent is locked or busy during installation, a **Reboot** dialog box appears to prompt you to reboot your system.

Viewing registry information

Applications running in Windows platforms rely on the registry for startup information, such as color, font, and key bindings.

Note: Proceed with caution when viewing registry information. Any change you make to the registry, accidentally or intentionally, could have an unexpected and potentially adverse affect on your application.

The OpenEdge installation adds the required `progress.ini` file information into the registry as entries under the following keys:

```
HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\10.1B
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\10.1B
```

Note: There is different `progress.ini` file information in each key.

The installation also automatically adds entries when you install an ODBC driver. For example, if you install the DataServer for ODBC, the following entries appear:

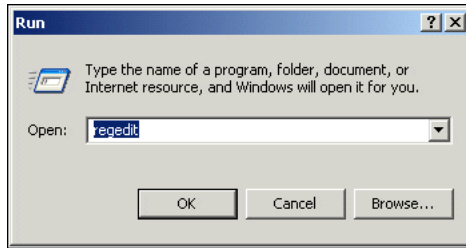
```
HKEY_CURRENT_USER\SOFTWARE\ODBC
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\OE Personal RDBMS\10.1B
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\Progress ODBC\10.1B
```

To add `progress.ini` file information into the registry, run the `ini2reg` utility. (The `ini2reg` updates the `HKEY_CURRENT_USERS` key.)



To review registry information:

1. On your desktop, choose **Start**→**Run**. The **Run** dialog box appears.
2. Type **regedit** (or **regedit32**) in **Open**, as shown:



3. Choose **OK**. A directory tree appears.
4. Double-click the `HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\10.1B` key or the `HKEY_LOCAL_MACHINE\SOFTWARE\PSC\OpenEdge\10.1B` key to view its contents.

Downloading executables for heterogeneous environments

The distributed architecture of OpenEdge allows you to optimize your hardware and network resources by installing components across networked machines, specifically when you are installing an OpenEdge Application Server component. Although some of these products' components must reside together on the same machine, you can, in some cases, distribute components to different machines, even if the machines run on different platforms. For example, you can install a WebSpeed Messenger or the NameServer in a Windows 2000 platform and install a WebSpeed Transaction Server on a UNIX platform.

The following list identifies executables that you can download for a platform other than Windows:

- WebSpeed Messenger.
- NameServer.
- Secure AppServer Internet Adapter (AIA).
- Web Services Adapter.

- OpenEdge Adapter for Sonic ESB.
- Promsgs (OpenEdge Messages).

The executables can be downloaded free of charge from Progress Software's Download Center at <http://www.progress.com/esd>.

OpenEdge Silent installation overview

An interactive installation prompts you for input and records your values in a series of dialog boxes. The Installation Program immediately uses this data to setup your OpenEdge products.

In contrast, a Silent installation is a two-step process:

- Data entered during the interactive installation process is recorded, typically in an `.ini` file. The OpenEdge installation automatically creates a `response.ini` file during the interactive installation process. Although you can create your own `.ini` file, the automatically-generated `response.ini` file is a reliable data input to perform a Silent installation.

This section focuses primarily on using the `response.ini` file because this data input does not require you to perform any additional file-related tasks. Optional response file-related activities, such as editing a response file, are presented later in this section.

- The installation data captured in an `.ini` file is read programmatically to install the products through a batch, or silent, mechanism at any time. Complete and custom installation support the Silent installation feature.

Note: If you plan to distribute a Silent installation that includes OpenEdge products that require Microsoft .NET Framework as part of the installation process, verify that the .NET Framework software is available on the system to which you are installing **before** you initiate the installation. Otherwise the Silent installation process will terminate.

- Selecting which `.ini` file to use to enter your installation values.
- Entering the command to start the Silent installation.
- Checking the status of the installation log.

Data input options for a Silent installation

Table 4–3 identifies and briefly describes the two types of data inputs you can use to perform a Silent installation.

Table 4–3: Data input options for a Silent installation

Data input options	Description
Automatically generated response.ini file	<p>An OpenEdge 10.1B interactive installation automatically creates a response.ini file that contains the installation values as you originally entered them in fields on the dialog boxes. It is stored in the install subdirectory in your installation directory, <i>OpenEdge-install-dir</i>. The file is immediately available for you to play back to start a Silent installation.</p> <p>See the “Understanding the Response.ini file contents” section on page 4–33 for more information and an excerpt of the response.ini file.</p>
User-initiated programmatic method	<p>Provides Application Partners (APs) a streamlined approach to integrate the OpenEdge installer into an application installer. Using this method, an AP can access the automatically generated response.ini file to programmatically create an OpenEdge installation response file; when the AP’s application is installed on a customer site, the OpenEdge installation information is read from the response file, enabling the customized install to be performed silently.</p> <p>For more information about this optional activity, see the “Creating data input option” section on page 4–40.</p>

Note: You can choose to edit the response file. However, keep in mind that any modifications to the automatically- or programmatically-generated response file can be time consuming and error prone.

Understanding the Response.ini file contents

The data captured in the `response.ini` file provides a detailed, reliable snapshot of the installation choices made during an interactive installation. As noted in [Table 4-3](#), the `response.ini` file is stored in your installation directory, *OpenEdge-install-dir*.

The `response.ini` file includes:

- A header version number and application details.
- Section labels defined by brackets for easy referencing.
- Each dialog box comment section identified with the label `DESCRIPTION` and the specific dialog box title.
- Easy-to-read descriptions of the fields on each dialog box.
- Only the values captured during the interactive install are stored in the `response.ini` file; there is no extraneous content.
- Dialog boxes that appear in the same order as presented in the online installation.
- A complete list of products installed.

The initial `response.ini` file is created when you run the Silent installation; it is never overwritten. If you re-run the Silent installation to add products to an existing 10.1B installation, a new unique `response.ini` file is created. It is identified as `response.ini.1`; any subsequent Silent installations will generate `response.ini.2`, `response.ini.3`, and so forth. These files will be saved to your installation directory.

Response.ini sample excerpt

The following example shows an excerpt from the automatically-generated response.ini file:

response.ini

(1 of 3)

```
[InstallShield Information]
Version=7.1.100.1242

[Application]
Name=OpenEdge
Version=10.1B
Company=Progress Software
File=Response File
;
; DESCRIPTION of Welcome Dialog
; Result - is used as the return code for this section. Only a value of 1 is
; acceptable.
[Welcome Dialog]
Result=1

; DESCRIPTION of Serial Number And Control Codes Dialog
; ProductCount - the number of products being installed.
; SerialNumber - the serial number of the product being installed.
; ControlNumber_1-0 - the first control code for the product being installed,
; where -0 indicates the first product.
; ControlNumber_2-0 - the second control code for the product being installed,
; where -0 indicates the first product.
; ControlNumber_3-0 - the third control code for the product being installed,
; where -0 indicates the first product.
[Serial Number And Control Codes Dialog]
ProductCount=7
SerialNumber-0=XXXXXXXXX
ControlNumber_1-0=xxxxxx
ControlNumber_2-0=xxxxxx
ControlNumber_3-0=xxxxxx
SerialNumber-1=XXXXXXXXXX
ControlNumber_1-1=xxxxxx
ControlNumber_2-1=xxxxxx
ControlNumber_3-1=xxxxxx
SerialNumber-2=XXXXXXXXXX
UseColorEditor=yes
Result=1
.
.
; DESCRIPTION of License Dialog
; AcceptLicenseAgreement - a value of 1 indicates the license agreement has
; been accepted, any other value indicates non-acceptance.
; Result - is used as the return code for this section. Only a value of 1 is
; acceptable.
.
.
```

response.ini

(2 of 3)

```

.
.
.
; DESCRIPTION of UserInstallationType Dialog
;
; InstallationType - identifies the type of product installation you plan to
perform. The valid values are complete and custom.
; - A Complete installation installs all mandatory, recommended, and optional
components and subcomponents of the OpenEdge products you are installing.
; - A Custom installation provides advanced users the opportunity to
selectively install recommended and optional components and subcomponents on
a product-by-product basis.
; Result - is used as the return code for this section. Only a value of 1 is
acceptable.
;
[UserInstallationType Dialog]
InstallationType=complete
Result=1

;
; DESCRIPTION of Configuring / Installing Components Dialog
;
; ConfigureSonicESBAdapter - used to indicate whether or not you want to
manually configure the OpenEdge Adapter for Sonic ESB, or use default values.

; - a value of 0 indicates default values will be used.
; - a value of 1 indicates the SonicEsbProperties dialog will be used to
set values.

; ConfigureWebSpeedMessenger - used to indicate whether or not you want to
manually configure WebSpeed Messenger, or use default values.

; - a value of 0 indicates default values will be used.
; - a value of 1 indicates the WebServer Type dialog will be used to set
values.
; InstallingProgressDynamics - used to indicate whether or not you want to
install Progress Dynamics files.

; - a value of 0 indicates Progress Dynamics files will NOT be installed.
; - a value of 1 indicates Progress Dynamics files WILL be installed.
; Result - is used as the return code for this section. Only a value of 1 is
acceptable.
;
[Configuring / Installing Components Dialog]
ConfigureSonicESBAdapter=1
ConfigureWebSpeedMessenger=1
InstallingProgressDynamics=1
Result=1

;

```

response.ini

(3 of 3)

```
.
.
; DESCRIPTION of Select Program Folder Dialog
;
; ShortcutFolder - the program folder in which your OpenEdge program shortcuts
will appear.
; Result - is used as the return code for this section. Only a value of 1 is
acceptable.
;
[Select Program Folder Dialog]
ShortcutFolder=OpenEdge
Result=1

;
; DESCRIPTION of SelectServerEngine Dialog
;
; UseSqlServerEngine - valid values are 0 and 1.
; 0 - indicates that the SQL Database Engine is to not be installed.
; 1 - indicates that the SQL Database Engine is to be installed.
; Result - is used as the return code for this section. Only a value of 1 is
acceptable.
.
.
;
[AdminServer Authorization Options Dialog]
GroupList=PSCAdmin
RequireUsernameAndPassword=0
EnableGroupChecking=0
Result=1
;
; DESCRIPTION of Summary Dialog
;
; Result - is used as the return code for this section. Only a value of 1 is
acceptable.
;
[Summary Dialog]
Result=1

[Installed Products]
ProductCount=7
Product 105=OE Enterprise RDBMS

Product 106=OE DataServer for Oracle
Product 108=OE DataServer MS SQL Svr.
;
```

Running the Silent installation

The command you use to initiate, or play back, the response file is the same regardless of which data input option you choose. The OpenEdge Silent utility runs without intervention after you enter the command to start the process.

The syntax for running the OpenEdge Installation utility in silent mode is:

Syntax

```
<path-to-install-media>\setup.exe -psc_s [-notify]  
-psc_f1=<path>\<response-file-name> [-psc_f2=<path>\<logfile-name>]
```

Note: Do not leave a space between command line entries and options. Also, neither command line entries nor options are case sensitive.

<path-to-install-media>\setup.exe

Command to run an OpenEdge installation. The *<path-to-install-media>* indicates that you can run the installation from CD, DVD, or download the executable from the Progress Download Center. The *setup.exe* identifies the specific OpenEdge installation executable.

-psc_s

Indicates a Silent installation.

-notify

Indicates that the installation dialog boxes that display will contain details about the current installation phase and percent complete. This element is supported for backward compatibility only.

However, the preferred method is to set up your application installation program to poll the log file for status of the installation process. You can programmatically set up a query, checking the Runtime Status and Result Code values in the log file. See the [“Checking the status of the Silent installation log file”](#) section on page 4–38 to review the contents of a sample oesetup.log file that contains Runtime Status and Result Code details.

`-psc_f1=<path><response-file-name>`

Indicates that the response file will be created, and specifies the pathname and filename of the file. By default, the install will look for the response file `oesetup.ini` in the same directory as `setup.exe` is located.

`-psc_f2=<path><logfile-name>`

Indicates that an installation log file will be created, and specifies the pathname and filename of the installation log file. If no filename is specified, the OpenEdge Installation Utility provides the default log filename `oesetup.log`.

If you do not specify a value for `<path>`, the Installation Utility writes this file to the Windows directory.

Example

The following syntax shows a typical Silent installation command:

```
\\cd-server\OpenEdge\setup.exe -psc_s  
-psc_f1=C:\SilentInstalls\oesetup.ini -psc_f2C:\SilentInstalls\  
oesetup.log
```

Checking the status of the Silent installation log file

The Silent Installation process automatically generates a log file, in which all messages—error and successful installation—as reported.

A log file, `oesetup.log`, is automatically generated for the OpenEdge install. The data captured in this log is useful when performing a silent installation. It contains status information that you can query at run time. The `oesetup.log` file can also be used to debug a silent installation.

An initial `oesetup.log` file is created when you install. Any time you re-run the Silent install, subsequent, unique `oesetup.log` files are automatically created and saved as `oesetup.log.1`, `oesetup.log.2`, `oesetup.log.3`, and so forth. In all cases, by default the file is located in `C:\Windows`.

An excerpt from an oesetup.log file follows:

oesetup.log

```
[InstallShield]
Version=7.1.100.1242
[Application]
Name=OpenEdge
Version=10.1B
Company=Progress Software
[CompletedEvents]
Event1=[9-22-2006 11:13:34] The Setup Utility has checked the system
requirements.
Event2=[9-22-2006 12:01:13] The Setup Utility has copied the support files.
Event3=[9-22-2006 12:01:13] The Setup Utility has created the config file.
Event4=[9-22-2006 12:02:17] The Setup Utility is extracting archives
Event5=[9-22-2006 12:23:56] The Setup Utility has extracted archives
Event6=[9-22-2006 12:24:01] The Setup Utility has installed system files.
Event7=[9-22-2006 12:24:22] The Setup Utility is configuring files.
Event8=[9-22-2006 12:24:23] The Setup Utility has configured language.
.
.
[RuntimeStatus]
Progress=99
[SystemFiles]
File1=[9-22-2006 12:23:56] C:\WINDOWS\system32\stdole2.tlb was successfully
installed.
File2=[9-22-2006 12:23:56] C:\WINDOWS\system32\sstree.ocx was successfully
installed.
File3=[9-22-2006 12:23:56] C:\WINDOWS\system32\pstimer.ocx was successfully
installed.
File4=[9-22-2006 12:23:56] C:\WINDOWS\system32\pstimer.chm was successfully
installed.
.
.
Shortcut32=[9-22-2006 12:32:33] Uninstall OpenEdge
[Reboot]
Required=no
[ResponseResult]
ResultCode=0
ResultDescription=Installation completed successfully.
```

Optional data input activities

The following optional activities are also supported when you are performing a Silent installation. However, keep in mind that creating the response file manually or editing the response file is the more time-consuming and potentially error-prone approach than creating it using the automatic response file method described in the [“Understanding the Response.ini file contents”](#) section on page 4–33.

Creating data input option

You can choose to record a separate response file any time you perform an interactive installation. If you do not specify a filename for the response file that you create, the install provides the filename `oe-response.ini` and stores in `C:\Windows\oe-response.ini`. The format and structure of any data input option is identical to that which is presented in the automatically-generated `response.ini` file. See the [“Response.ini sample excerpt”](#) section on page 4–34 to review an excerpt of the file’s content.

The syntax for initiating the user-defined response file is:

```
<path-to-install-media>\setup.exe  
-psc_r [-psc_f1=<path>\response-file-name]
```

Note: Do not leave a space between command line entries and options. Also, neither command line entries nor options are case sensitive.

`<path-to-install-media>\setup.exe`

Command to run an OpenEdge installation. The `<path-to-install-media>` indicates that you can run the installation from CD, DVD, or download the executable from the Progress Download Center. The `setup.exe` identifies the specific OpenEdge installation executable.

`-psc_r`

Indicates that the install is in record mode.

`-psc_fl=<path><response-file-name>`

Indicates that the response file will be created, and specifies the pathname and filename of the file. By default, the install will look for the response file `oesetup.ini` in the same directory as `setup.exe` is located.

Manually modifying data input option

You can edit any response file, whether it is automatically generated or one you create. Although all sections of the response file are required, you do not need to add each of these required sections in the order presented. The installer only retrieves the specific data it needs regardless of where the information is located in the response file.

Performing postinstallation tasks

Before you run OpenEdge, there are some postinstallation tasks you might need to complete, depending on your application needs and goals:

- **Completing the Progress Dynamics Configuration Utility (DCU)** — The DCU wizard guides you through the setup steps to install Progress Dynamics. For the procedures to complete the DCU, see the [“Completing the DCU wizard”](#) section on page 4–7.
- **Completing third-party software installations** — If you installed products that required Microsoft .NET Framework, and you agreed for it to be installed, it will launch automatically. See the [“Required third-party applications”](#) section on page 1–10.
- **Set environment variables** — For more information on setting environment variables (including SQL), see [Chapter 7, “Working in the OpenEdge Environment in Windows.”](#)
- **Create customized executables** — To create customized product executables, see the information on building ABL executables in [OpenEdge Deployment: Managing ABL Applications](#). Creating executables might be required for certain product configurations.
- **Re-apply properties file details (if needed)** — See the [“OpenEdge automatic save of Properties files”](#) section on page 3–17 for details.

- **Validate and populate group names created for the AdminServer security option** — The tasks used to verify that your groups were created, and to identify the specific members of these groups, are completed outside of the OpenEdge installation. The installation process only allows you to create the groups that you specify.

Note: For detailed information about how to verify that groups have been created, and how to access and set up group members for each group in a Windows platform, refer to your operating system-specific documentation. The criteria you use to set up users within each group is determined by your company.

Also, depending on your installation requirements, you might need to perform this task:

- **Edit files to point to a previously installed JDK** — If the required version of the Java Soft (InstallShield) JDK was installed on your system prior to the OpenEdge Version 10.1B installation and you choose to use this pre-existing JDK utility, you now, as a postinstallation task, must edit files tailored by the install to ensure that they point to this pre-existing JDK. Contact Progress Technical Support for assistance to perform this task.



To verify that your groups exist and to create users for each group after you have successfully installed OpenEdge:

1. From the **Control Panel**, choose **Administrative Tools**→**Computer Management**. The **Computer Management** dialog box appears.
2. Choose **System Tools** to display a list.
3. Select **Local Users and Groups**, and choose **Groups**.

Verify that the groups you created during the installation process are among those listed.

4. To populate a given group, double-click the group name and complete the **Users Properties** dialog box.

Uninstalling OpenEdge in Windows

When you delete files from the OpenEdge directory tree, you only partially remove an OpenEdge installation. By contrast, the Uninstall utility or the Remove Program utility removes all OpenEdge files as well as configuration information from the registry, which prevents conflict with subsequent OpenEdge installations. If for some reason the Uninstall utility cannot completely uninstall OpenEdge, you must manually remove the installation.

If you want to upgrade or remove an installation, choose one of the following:

- Run the Uninstall utility from the OpenEdge program group, as described in the “[Using the Uninstall or Add/Remove Programs utility](#)” section on page 4–43.
- Run the Add/Remove Programs utility from the Microsoft Windows Control Panel, as described in the “[Using the Uninstall or Add/Remove Programs utility](#)” section on page 4–43.
- Manually remove the installation, as described in the “[Manually removing OpenEdge](#)” section on page 4–46. For additional details, see the Progress Software Company Knowledge Center at <http://progress.atgnow.com/esprogress>.

Note: You can uninstall OpenEdge from a MetaFrame platform in the same way as from Windows operating systems. MetaFrame operates in a client/server environment and is not the same as a shared network environment. Therefore, you cannot share an OpenEdge installation on a network that operates on MetaFrame.

Using the Uninstall or Add/Remove Programs utility

You can run the Uninstall utility (or use the Add/Remove Programs utility located in the Windows Control Panel) to automatically remove OpenEdge from your system. Running the Uninstall or the Remove Program utility removes configuration information from the registry and prevents conflict with subsequent OpenEdge installations.

Caution: When uninstalling, do not delete any of the following Microsoft system files: asycfilt.dll, comctl32.ocx, ctl3d32.dll, mfc70.dll, mfc71.dll, mfcans32.dll, mscomctl.ocx, msvc70.dll, msxcp71.dll, msxcr.70.dll, msxcr71.dll, oc30.dll, oleaut32.dll, olepro32.dll, pdh.dll, psapi.dll, stdole2.tlb. These system files are common to other applications, and deleting them might adversely affect the operation of the other applications that use them. To avoid deleting these system files while running the Uninstall utility, answer **No** to the prompts at the end of the uninstall process.



To run the Uninstall utility:

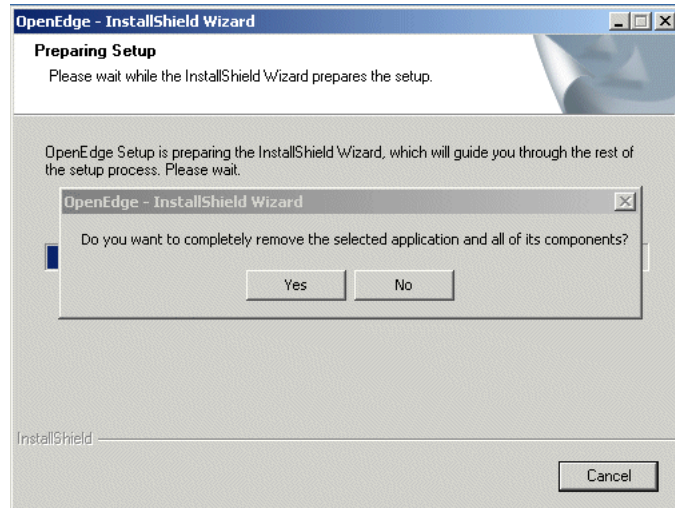
1. Log in under the same domain and user name you used when you installed OpenEdge.
2. Make sure that OpenEdge is not running in an open DOS window (or that the current directory is not any OpenEdge-related directory).
3. Stop all OpenEdge processes, including your Web server (for example, you might be using a Microsoft Web server or ISAPI-compatible, or Sun Web server or NSAPI-compatible Web server), and close all OpenEdge help files. Use the Task Manager to ensure that you stop all processes and close all help files.
4. Using the Progress Explorer, shut down all OpenEdge and WebSpeed services (brokers, NameServers, and database servers).
5. Make sure that your Web server, as identified in Step 3 of this procedure, is shut down.
6. From the Windows desktop, select **Start→Settings→Control Panel→Administrative Tools→Services**.

Highlight the **AdminService for OpenEdge 10.1B**, and choose **Stop**. Change the startup from **Automatic** to **Manual**, and choose **OK**.

Note: If these same services will be required for a new installation, be sure to note any configuration settings, agent parameters, etc.

7. Do one of the following tasks to uninstall:
 - a. Choose the **Uninstall** icon in the OpenEdge Program Group.
 - b. From the desktop, select **Start**→**Control Panel**→**Add or Remove Programs**. Select **OpenEdge** and choose **Change/Remove**.

The **InstallShield Wizard** appears:



8. Choose **Yes**.
9. Choose **OK** when the deletion process is complete. If the uninstall was successful, you have finished. However, if the uninstall failed or terminated abnormally during the process, you must manually remove the OpenEdge Uninstall folder. Refer to the [“Manually removing OpenEdge”](#) section on page 4–46 for the procedure to complete.

Manually removing OpenEdge

If you attempted to perform the uninstall procedure outlined in the “[Using the Uninstall or Add/Remove Programs utility](#)” section on page 4–43 and it failed, you must manually remove the OpenEdge Uninstall folder before reinstalling. The uninstall file records the initial installation and appends additional installations to the file.

This section provides guidelines to manually uninstalling the OpenEdge folder located at C:\Program Files\InstallShield Installation Information\{3D65631B-B94E-47C9-9AEA-E80AA431E841} and perform other steps related to this task.



To manually uninstall OpenEdge:

1. Log in under the same domain and user name you used when you installed OpenEdge.
2. Make sure that OpenEdge is not running in an open DOS window (or that the current directory is not any OpenEdge-related directory).
3. Stop all OpenEdge processes and close all OpenEdge help files. You can use the Task Manager to ensure that you stop all processes and close all help files.
4. Using the Progress Explorer, shut down all OpenEdge services (brokers, NameServers, and databases).
5. Shut down the AdminServer by following these steps:
 - a. From the desktop, select **Start→Control Panel→Administrative Tools→Services**.
 - b. Highlight the **AdminService for OpenEdge 10.1B**, and select **Stop**.
 - c. When the service stops, choose **Action→Properties**. The **AdminService** dialog box appears.
 - d. Change the **Startup type** from **Automatic** to **Manual**, and choose **OK**. (This step is necessary if you reboot your machine before completing the uninstall so that the AdminServer does not start up automatically.)

Note: If these same services will be required for a new installation, be sure to note any configuration settings, agent parameters, and so forth.

6. Shut down some services, as needed. Consider the following situations:
 - If you are using a Sun Web server (or NSAPI-compatible Web server) that uses the `wsnsa.dll`, you are not required to shut down a Windows service. You only have to shut down the Web server and the WebSpeed Transaction Server.
 - If you are using the Microsoft IIS Web server (or NSAPI compatible Web server) to employ the WebSpeed Messenger that uses the `wsisa.dll`, you must shut down the IIS Admin Service.
7. Remove the `C:\Program Files\InstallShield Installation Information\{3D65631B-B94E-47C9-9AEA-E80AA431E841}` directory.
8. Run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry as follows:
 - a. Remove the 10.1B keys that appear under the `HKEY_CURRENT_USER` location. If there is only one release number identified under `PSC`, delete the `PSC` key, as shown:

`HKEY_CURRENT_USER\SOFTWARE\PSC\Progress\release number`

- b. Remove the 10.1B keys that appear under the `HKEY_LOCAL_MACHINE` location. Check the release number identified under each product subfolder. If only one release number is identified as installed for all products, delete the `PSC` key, as shown:

`HKEY_LOCAL_MACHINE\SOFTWARE\PSC\product name(s)\release number(s)`

- c. Remove the 10.1B key that appears under the following `HKEY_LOCAL_MACHINE` location:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`

9. If you have installed an OpenEdge product for which any drivers have also been installed, run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry as follows:
- a. Go to `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI`.
 - b. Remove the following key(s) (and the values it contains), as needed:

DataDirect 4.20 32-BIT OpenEdge SQL v10.1B
OpenEdge 10.1B Informix Driver
OpenEdge 10.1B Informix Wire Protocol Driver
OpenEdge 10.1B SQL Server Wire Protocol Driver
OpenEdge 10.1B Sybase Wire Protocol Driver

10. Depending on the products you have installed, the following files might have been registered during the install and should be unregistered:

dzocx32.ocx	pstimer.ocx
dzstat32.ocx	prox.dll
duzocx32.ocx	sstree.ocx
cscomb32.ocx	cihttp.ocx
csspin32.ocx	cslist32.ocx

Note: The registered version of some of these files might not be under the OpenEdge installation directory. Also check the Windows `system` and `system32` directories for these files.

The following example shows how you can unregister one of these OCX files, which, in the example, is located in the Windows system32 directory:

```
OpenEdge-install-dir\bin\regsvr32.exe/u c:\windows\system32\cscomb32.ocx
```

Note: If the OpenEdge Uninstall utility does not complete successfully, you must perform all or some of [Step 11](#) through [Step 15](#) before the new installation can take place. For details, see the Progress Software Company Knowledge Center at <http://progress.atgnow.com/esprogress>.

11. Delete the OpenEdge program directory, including all of its subdirectories. The default OpenEdge directory is C:\Progress\OpenEdge.
12. Delete the OpenEdge folder/group from the Windows **Start** menu.
13. Shut down your Web server and delete the `cgiip.exe` and `wsisa.dll` files from the Web server `cgi-bin/scripts` directory.

Note: If you are uninstalling WebSpeed and using the Sun Web server that uses the `wsnsa.dll`, you must return the `obj.conf` file to its pre-WebSpeed state. If you are upgrading WebSpeed to the same directory, you need not modify the `obj.conf` file. However, if you intend to change the directory location, then you must modify the `obj.conf` file to reflect the correct location.

14. Depending on the installation options you chose (that is, Web server type, WebSpeed virtual directory, or having static HTML files copied to your Web server document root directory), you might need to perform one or more of the following steps:
 - a. Delete the WebSpeed directory from your Web server document root directory; for example, on MSIIS the default is: `\InetPub\wwwroot\webspeed`.
 - b. Delete any virtual directories defined for WebSpeed in your Web server.
15. Reboot your machine and install OpenEdge.

Sharing an OpenEdge installation on a network overview

You can use OpenEdge networking functionality and the Shared Network Installation Utility (NetSetup) to install a single copy of the OpenEdge Installation Program on a network-accessible drive (server) and enable multiple clients to access it.

Primary tasks

The primary tasks to share an OpenEdge installation on a network are:

- Performing an OpenEdge installation on a network server machine, using the OpenEdge Installation Program.

During the installation process, the Shared Network Installation (NetSetup) Utility—the component that allows each client machine to install the required software to access the network server machine—is installed on the server. In a Complete installation, the NetSetup component is automatically installed with all OpenEdge products. In a Custom installation, you must select the NetSetup component as an optional component. The NetSetup Utility also supports a silent installation option.

- Using the NetSetup Utility to update each client machine to enable it to access the network server's installation copy.

The NetSetup Utility ensures that all the system files, icons, and registry entries needed to launch the OpenEdge products locally are set up on each client machine. The NetSetup Utility is comprised of one dialog box, the **Destination and Work Paths** dialog, that you run on the client.

The details to address the tasks previously listed and other related activities are described in the following sections:

- [Networking overview.](#)
- [Determining a shared network to clients connection.](#)
- [Setting up the shared network.](#)
- [Running the Shared Network Installation Utility to set up a client connection.](#)
- [Reviewing local intranet security settings.](#)

- [Uninstalling the Shared Network Installation Utility.](#)
- [Running the Silent installation option for the Shared Network Installation Utility.](#)

Networking overview

This section provides some background information about the basic networking hardware needed to run OpenEdge in a network to client configuration.

A network typically consists of the following hardware elements:

- **Application workstation** — A computer on your network that executes the OpenEdge Client or single-user product. This software allows one or more users to access the database server machine.
- **Database server machine** — The OpenEdge database server machine is a computer on your network that executes the OpenEdge Server software. This software allows the database server machine to manage one or more OpenEdge databases.
- **Network file server machine** — The network file server is a computer that manages file sharing and system security, coordinates station-to-station communications, and controls any attached peripherals, such as printers, disk drives, and modems.

You can install the OpenEdge client software on each machine where it will run, or you can install one copy of each on the network file server. For more information on networking with OpenEdge, see the “[Working with Unified Brokers](#)” section on page 10–9.

Note: Progress Software Corporation (PSC) does not support installing one copy of the OpenEdge Application server products to be run by multiple machines because there is only one set of configuration files and conflicts would occur.

Determining a shared network to clients connection

You can use the following connection types to share OpenEdge installed on a network server with multiple client machines:

- Mapped drive.
- Uniform Naming conventions (UNC) pathnames.

Setting up the shared network

This section describes setting up the OpenEdge products on your network server.

Prerequisites

Note the following before you set up a shared network installation on your network server:

- Uninstall any existing OpenEdge or Progress product that is installed on client machines to which you will be installing. See the [“Uninstalling OpenEdge in Windows”](#) section on page 4–43.
- Review the OpenEdge installation tasks. See the [“OpenEdge Installation Prerequisites”](#) section on page 3–1 and the [“Installation overview”](#) section on page 4–2.
- Determine the destination location of your OpenEdge installation on the network. You will be prompted to enter it during the network installation, and when you use the NetSetup utility to install the connecting clients.



To perform a shared network installation on your network server:

1. Run the OpenEdge installation.
2. When the **Choose Destination And Working Path Directories** dialog box appears, make a note of the location you type in the **Destination Directory** field. You need this directory path information when you install on each client machine.
3. Complete the OpenEdge installation on the shared network.

Running the Shared Network Installation Utility to set up a client connection

The Shared Network Installation Utility (NetSetup) updates each client machine with all system files, icons, and registry entries needed to launch OpenEdge locally. Each client can then share the networked copy of OpenEdge.

The client machine in a NetSetup installation uses the OpenEdge installer located on the network server. The installer software enables you to locally launch NetSetup.

Note: Uninstall any existing 10.1B OpenEdge product that is currently installed on a client machine to which you are installing. See the [“Uninstalling OpenEdge in Windows”](#) section on page 4–43.



To run NetSetup on your client machine:

1. Choose **Start**→**Run**. The **Run** dialog box appears.
2. In **Open**, type one of the following supported connection options to connect the client machine to the shared network server:
 - a. To identify a mapped drive connection, type:

`drive:\destination_path\netsetup\setup.exe`

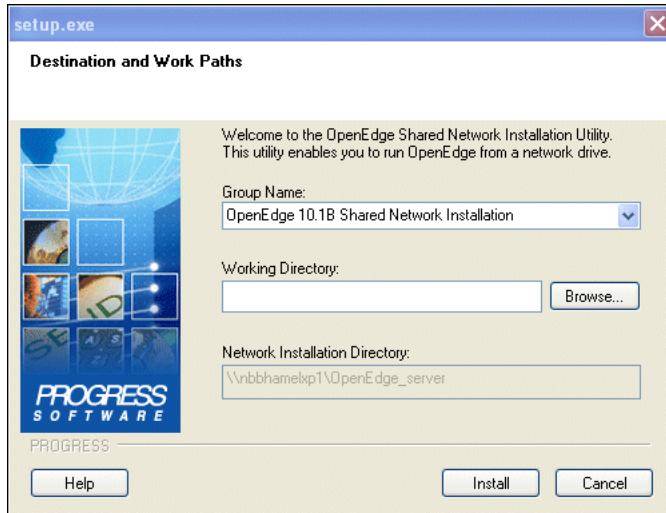
The *destination path* is the same path where OpenEdge is installed on the server machine.

- b. To identify the UNC pathname connection, type:

`\\servername\sharename\destination_path\netsetup\setup.exe`

The *destination path* is the same path where OpenEdge is installed on the server machine.

3. Choose **OK**. The **Destination and Work Paths** dialog box appears:



4. Accept or change the **program group** name that appears in the **Group Name** field. The **Group Name** value identifies the menu option label that appears on the client machine. When you select this name from **All Programs**, you can access the OpenEdge installation that resides on the network server.

Note: If the group name does not already exist, the NetSetup utility adds the group name to **All Programs**.

5. Type the absolute path or browse to find the file to identify as the client-based working directory in the **Working Directory** field. The **Working Directory** is a local folder in which OpenEdge places the databases you create on the client.
6. Review the pathname information that appears in the **Network Installation Directory** field. This pathname identifies where OpenEdge is installed on the network server.

Note: The Network Installation Directory field always appears grayed out, confirming that the information that appears in this field cannot be changed. The pathname that appears in this field identifies two pieces of information: where OpenEdge is installed on the network server and the type of connection that you are using to share the network installation (that is, mapped drive or UNC pathname).

7. Choose **Install**. Select the Group Name you defined from **All Programs** to access the OpenEdge installation from the network serve.

Note: If you change the original installation on the network, and the installation includes additional shortcuts supported by the Netsetup Utility, you must uninstall and reinstall the NetSetup Utility on the client to ensure that the shortcuts are available on the client machine.

Shortcuts

Table 4–4 shows all of the OpenEdge product-specific shortcuts that can be potentially available on a shared network server:

Table 4–4: Available Network Server and Client Shortcuts

AppBuilder	Desktop
Application Compiler	Help
Audit Policy Maintenance	Proenv
Character Client	Progress Explorer Tool
Client	Proxy Generator
Config	Release Notes
Data Administration	RESULTS
Data Dictionary	Translation Manager
Debugger	Visual Translator

Keep in mind that the specific shortcuts available to a shared network server and its client machines will vary, depending on the actual OpenEdge products installed on the network

Reviewing local intranet security settings

After you have performed an OpenEdge Shared Network Installation, you might find some products are not be accessible when you attempt to launch them from the Program Group. For example, a managed application such as Proxy Generator might not launch properly when selected from the Program Group icon; it is possible that the local Intranet security setting might be set too high.

Review and, if needed, change the current .NET security configuration to enable Proxy Generator to run successfully. To do so, you must run the .NET Framework Wizards to reset security options.



To change the .NET security configuration to ensure that you have appropriate settings to run Proxy Generator:

1. From the desktop, choose **Start→Control Panel→Administrative Tools→Microsoft .NET Framework Wizards**.
2. Choose **Adjust .net Security** and choose **Next**.
3. Choose **Make changes to this computer** (default).
4. Choose **Local Intranet** and choose **Next**.
5. Move the slider bar to **Full Trust**. This setting ensures that you can access and use Proxy Generator.
6. Choose **Next** and choose **Finish**.

Note: You must perform the procedure on the server and on each client system from which you intend to use Proxy Generator.

Uninstalling the Shared Network Installation Utility

You can use an uninstall utility to uninstall the NetSetup Utility from a client machine that is currently connected to an OpenEdge shared network installation. All the products that you previously installed for this OpenEdge release are removed. This procedure must be done for each client machine you intend to uninstall.



To uninstall the client machine from a Shared Network installation:

1. From the desktop, choose **Start→Control Panel→Add Or Remove Programs**.
2. From the list of installed programs, select the **OpenEdge 10.1B Shared Network Installation**. Choose **Change/Remove**. A confirmational dialog box appears.

Note: Remove client files first, then uninstall the server to ensure that the shared network installation is properly uninstalled.

3. Choose **Yes** to confirm that you want to delete the OpenEdge 10.1B Shared Network Installation from your client machine. The **Remove Programs From Your Computer** dialog box appears.
4. Choose **OK** to exit the Uninstall utility from the client machine.

Note: When the usage count on a shared system file reaches 0, a **Shared File warning** dialog box appears. Follow the instructions in the dialog box.

Running the Silent installation option for the Shared Network Installation Utility

The Shared Network Installation Utility (NetSetup) supports a Silent installation process. The process is comparable to performing a Silent installation of the OpenEdge installation: data entered during a NetSetup installation is recorded and played back at a later date to initiate the installation silently.

To perform a NetSetup Silent installation, however, you must create your own response file. Unlike the OpenEdge installation, there is no automatically-generated `response.ini` created during a NetSetup installation.

The information in this section describes:

- [Creating a user-defined response file.](#)
- [Executing NetSetup with the silent installation option.](#)
- [Reviewing details about the `PscNetSetupMsg.log` file.](#)

Creating a user-defined response file

Before you can run the NetSetup utility in silent installation mode, you must create the user-response file. This file records the values that the NetSetup utility needs to successfully complete the silent installation process. This section describes how to create a response file using the interactive method.

To create this file, you must perform an initial interactive installation, providing the required values.

**To create the user-defined response file using the interactive installation mode:**

1. Enter the following command on the command line:

```
drive:\destination_path\netsetup\setup.exe -r  
[-f1C:\<path-to-file>\response-file]
```

setup.exe

The command to run the NetSetup program interactively.

-r parameter

The Installshield mechanism to create the response file using the interactive method and to identify it as setup.iss. The setup.iss file is a text file that is modifiable. The -r parameter is the recommended method to ensure that you create a complete response file.

-f1<path>\<response-file-name>

Specifies the name of the response file. By default, the install will look for the file setup.ini in the same directory as setup.exe is located.

2. Press **ENTER**. NetSetup runs interactively.

When you type values through the keyboard, NetSetup simultaneously creates the response file, to record particular values that you enter. Values specific to your installation are read and stored in the response file.

The following example shows the typical contents of a sample response file, `setup.iss`:

setup.iss

```
[Silent]
Version=v7.00
File=Response File
[File Transfer]
OverwrittenReadOnly=NoToAll
[{874D5CE4-F913-4D5B-A6D4-CC129785B5C8}]-DlgOrder]
Dlg0={874D5CE4-F913-4D5B-A6D4-CC129785B5C8}-DLG_SHARED_INSTALL-0
Count=2
Dlg1={874D5CE4-F913-4D5B-A6D4-CC129785B5C8}-MessageBox-0
[{874D5CE4-F913-4D5B-A6D4-CC129785B5C8}]-DLG_SHARED_INSTALL-0]
ProgramFolder=OpenEdge 10.1B1B Shared Network Installation
WorkingDir=C:\OpenEdge\NetinstWrk
[Application]
Name=OpenEdge Shared Network Install Utility
Version=10.1B
Company=PSC
Lang=0009
[{874D5CE4-F913-4D5B-A6D4-CC129785B5C8}]-MessageBox-0]
Result=1
```

The values entered for the **Program Folder** and **WorkingDir** during the interactive installation are recorded in the response file. The shortcut information identified in the **Program Folder** and the user's work files identified in the **WorkingDir** are read during a silent installation.

Note: The **OverwrittenReadOnly** option ensures that the **Read Only File** dialog box is suppressed during a silent setup.

Executing NetSetup with the silent installation option

Once the response file exists, the installation process using the silent mode can be initiated.

To initiate NetSetup with the silent installation option, enter the following command on the command line to run NetSetup in silence:

```
drive:\destination path\netsetup\setup.exe -psclog[C:\<path-to-file>] -s
[-f1C:\<path-to-file>\response-file]
```

`drive:\destination path\netsetup`

The path to where the NetSetup utility resides on the server in the OpenEdge product file structure.

`setup.exe`

The command to run the NetSetup program.

`-psclog [C:\<path-to-file>]`

Required parameter to run NetSetup in silent installation mode. This parameter is also optionally used to identify a path to a log file that contains information about the status of the silent installation.

The log file created by the installation program is called `PscNetSetupMsg.log`.

`-s`

Required parameter to run an installation without requiring user interaction. This parameter is executed with the `setup.exe` to run a silent installation.

`-f1<path>\<response-file-name>`

Specifies the name of the response file. By default, the install will look for the file `setup.iss` in the same directory as `setup.exe` is located.

The following example shows the typical contents of the `PscNetupMsg.log` file:

```
[Progress NetSetup Messages]
Type=INFORMATION
Date=6-23-2005
Time=10:02:46
File=setup.rul
Line=987
Message=Setup is complete.  You may run the installed program.
=====
Type=INFORMATION
Date=6-23-2005
Time=10:02:46
File=setup.rul
Line=377
Message=Completed Successfully.
=====
```

Reviewing details about the PscNetSetupMsg.log file

Review the following points concerning the file `PscNetSetupMsg.log`:

- You must supply the absolute path to the log file because the installation program is shared by other users. Without this explicit path information, the log file might be incorrectly written to by more than one instance of the `setup.exe` file which would corrupt the file.
- The `PscNetSetupMsg.log` file automatically logs all error messages and all successful installation messages as reported by the installation code.

Performing an OpenEdge Installation on UNIX

This chapter contains the instructions for installing OpenEdge on UNIX, as outlined in the following sections:

- [Installation overview](#)
- [Additional product installation activities](#)
- [OpenEdge Silent installation overview](#)
- [Performing postinstallation tasks](#)
- [Uninstalling OpenEdge on UNIX and Linux operating systems](#)
- [Uninstalling OpenEdge Adapter for SonicMQ and OpenEdge Adapter for SonicESB](#)

Installation overview

After you have addressed all the topics presented in the “[Tasks overview](#)” section on page 3–2, you are prepared to perform the OpenEdge installation on either a UNIX or Linux platform.

Loading the installation media

Perform the following procedure to begin the installation process.



To initiate the Installation Utility to install OpenEdge products:

1. Obtain a copy of the completed *Preinstallation Checklist for UNIX*. You might also want to have the other installation-related documents highlighted in [Table 3–1](#) of the “[Gathering information to plan your installation](#)” section on page 3–3 available.

Note: When you install a client networking license, the ADM2 directory is not installed in the *OpenEdge-install-dir*/GUI directory. This r-code is considered part of your application and should be deployed as a module of your application.

2. Close all other applications before beginning the installation process.

Other applications or tasks might interfere with the installation or use files that OpenEdge needs to complete the installation. Shut down any processes where the executable itself, or a file used by the executable, is located in the directory where you intend to install OpenEdge.

3. Log in as root. If you do not know the root password for your machine, check with your system administrator.

4. Install the installation program from the installation medium you are using, as shown in the following table:

For this installation medium . . .	Do the following . . .
CD	Insert it into the CD drive.
DVD	Insert it in the DVD player and navigate the directory structure, locating the specific platform directory to which you intend to install. ¹
Electronic Software Distribution (ESD) download	Navigate to the software image you intend to download from the Progress Software Download Center. ²

1. The installation DVD contains subdirectories for all Windows and UNIX platforms that OpenEdge 10.1B supports. However, only the specific platform and type to which your license pertains is installable.
2. The Progress Software Download Center is available at <http://www.progress.com/esd>. Access to Progress software products and updates at this Web site requires a valid account.

Note: There is an automount daemon on the Solaris platform that mounts the CD within approximately five seconds. If however, the automount does not occur on your Solaris machine, go to [Step 5](#) in this procedure.

5. Enter your platform-specific mount command (where *device-name* is the device you are using for the installation and *mount-point* is the mount-point directory). The following table lists the mount commands for each supported platform:

Operating system	Mount command
HP-UX (PA RISC) (32-bit and 64-bit)	mount -F cdfs -r -o cdcase <i>device-name</i> <i>mount-point</i> For example: mount -F cdfs -r -o cdcase /dev/dsk/c0t2d0 /cdrom
HP-UX ITANIUM (IA 64)	mount -F cdfs -r -o cdcase <i>device-name</i> <i>mount-point</i> For example: mount -F cdfs -r -o cdcase /dev/dsk/c0t2d0 /cdrom
Tru64 UNIX 5.1B (64-bit)	mount -t cdfs -o noversion <i>device-name</i> <i>mount-point</i> For example: mount -t cdfs -o noversion /dev/rz3c /cdrom
IBM AIX 5.2 (32-bit and 64-bit)	mount -v cdrfs -r <i>device-name</i> <i>mount-point</i> For example: mount -v cdrfs -r /dev/cd0 /cdrom
Sun Solaris SPARC (32-bit and 64-bit)	mount -F hsfs -o ro,nrr -r <i>device-name</i> <i>mount-point</i> For example: mount -F hsfs -o ro,nrr -r /dev/dsk/c0t4d0s0 /cdrom
Red Hat Enterprise Linux 2.1	mount -t iso9660 <i>device-name</i> <i>mount-point</i> For example: mount -t iso9660 /dev/cdrom /cdrom
Unixware	mount -F cdfs -r <i>device-name</i> <i>mount-point</i> For example: mount -F cdfs -o /dev/cdrom/cdrom1 /cdrom

6. Enter the following install command:

`mount-point/proinst`

Note: You cannot run `proinst` if you are in the *mount-point* directory or the intended installation directory.

7. Proceed as shown in the following table:

If the JVM is . . .	Then . . .	And . . .
Found to be installed on your platform.	The Welcome dialog box appears.	Proceed with the installation.
Not found to be installed on your platform.	The Installation Utility searches your \$PATH for it.	If the JVM is then found in the \$PATH, the Welcome dialog box appears.
Not found to be installed on your platform.	The JVM has not been detected Warning message appears.	The installation continues, but you can only install products that do not require a JVM.
Not found to be installed on your platform.	The JVM version does not match the version supported by OpenEdge Warning message appears.	You can choose to continue with the installation whether or not you have the supported JVM version on your system. However, Progress Software Corporation recommends that you install the supported JVM version to ensure full functionality. ¹

1. If you are performing a batch installation, you can add an entry to the .ini file to allow batch installs to override this warning. See the [“OpenEdge Silent installation overview”](#) section on page 5–13 for more information.

The **Welcome** dialog box appears:



8. Proceed to the [“Performing the installation”](#) section on page 5–6.

Performing the installation

Once you have loaded the installation program from your installation medium and the **Welcome** dialog box appears, you are ready to perform the online tasks required to install OpenEdge.

Refer to [Table 3–1](#) for the documents you should reference during the installation to help you perform the online OpenEdge installation.

Navigating though the Installation Utility

The Installation Utility is designed to programmatically present the dialog boxes for which you need to enter data, according to the products you are installing and the type of installation you choose to perform. Record your input on each dialog box and advance to the next dialog box at your own pace. The specific controls you use to advance to the next dialog box or return to a previous dialog box are identified on each dialog box. Highlight a menu option using the **SPACEBAR** key, the **TAB** key, the **CURSOR** keys, or the accelerator keys that are highlighted in each selection on the dialog box.

You can generally use the **Cancel** control to toggle back to a previous dialog box to review and/or update your choices to date. **Cancel** also allows you to quit the installation at any time before you commit to your selections. You will also be given the option to not install any installation files at this time, and you can begin the installation process again at a later point.

Some dialog boxes also have unique buttons that allow you to complete a procedure or reset default values.

Accessing online help topics

Refer to the online installation help which contains a help topic for each installation dialog box. Access the online help according to the method identified on each dialog box; generally, you will enter the control-key sequence, or highlight the **Menu** option and press **ENTER**. Scroll in the help to view all of the dialog box explanatory and procedural details; press **ESC-ESC** to exit the help topic and return to the Installation Utility.

Installation-related messages

During the installation, additional questions, messages, or information related to certain dialog boxes might appear. Follow the instructions as presented.

Committing your installation choices

Once you are satisfied with all your selections, you can review a comprehensive list of your installation choices and commit them to be installed from the **Summary** dialog box.

Finishing the installation

You are now ready to understand, review, and make changes to, the user environment to run OpenEdge. Note the following:

- If you have installed an SQL product, you must set your environment variables. For more information on setting environment variables, either to support an SQL product or to customize the variables to your own preferences, see [Chapter 8, “Working in the OpenEdge Environment on UNIX.”](#)
- To create customized product executables, see the information on building ABL executables in *OpenEdge Deployment: Managing ABL Applications*. Creating executables might be required for certain product configuration.
- Address any other postinstallation tasks discussed in the [“Performing postinstallation tasks”](#) section on page 5–22.

Additional product installation activities

This section highlights the following additional product-related activities you might want to perform:

- [Installing additional products.](#)
- [Adding components to previously installed products.](#)
- [Downloading executables for heterogeneous environments.](#)

Installing additional products

Once you have successfully installed OpenEdge, you can elect to add additional products to your current installation.

To initiate this process, you must re-load your installation media. Perform the steps outlined in the [“Loading the installation media”](#) section on page 5–2. Then perform the steps outlined in the following instructions.

Note: When you add products to an existing installation, you can use the Installation Utility in batch mode as long as you are performing a Complete installation of the products you are adding. For more information about a batch installation, see the [“OpenEdge Silent installation overview”](#) section on page 5–13.



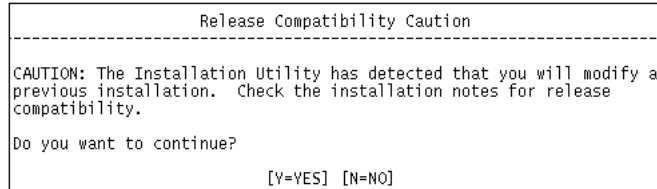
To install additional products when the Welcome dialog box appears:

1. Press **RETURN** to continue. The **Serial & Control Numbers** dialog box appears.
2. Enter only the control numbers for the products you are adding to the list of previously installed products.
3. When you are done, press **CTRL+E**. The **Done Configuration Data Confirmation** dialog box appears.
4. Press **Y** to continue (or press **N** to add more products). The **Type Device and Destination** dialog box appears.
5. Choose **Select the Destination Pathname**, and type the path of the initial installation.
6. Press **RETURN**. The **Destination Pathname Exists** dialog box appears:

Destination Pathname Exists	
Select an alternate destination path	
Erase the current destination path	
Install the OpenEdge products in the pre-existing destination path	
Help	

7. Choose **Install the OpenEdge products in the preexisting destination path** and press **RETURN** to continue with the installation.

If you install products that affect your previously installed products, you might see the following caution message:



8. Choose **Yes** to continue with the installation.

The installation program adds your OpenEdge products to your directories automatically.

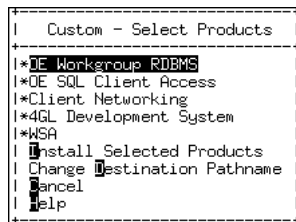
Adding components to previously installed products

You can add components and subcomponents to existing OpenEdge installations without having to enter any data other than the required components or subcomponents.



To add components or subcomponents using the Add feature:

1. At the command line, type the shell script *destination-path/proaddcomp* to run the add feature. The **Select Products** dialog box appears:



All previously installed products appear on this list. The **Select Products** dialog box allows you to select and deselect OpenEdge products for which you want to add components or subcomponents.

2. Select or deselect a product by highlighting the product and pressing **RETURN**. An asterisk (*) indicates that a product is selected. If you want to select the first product on the list, you must first press **RETURN** to deselect the product and then press **RETURN** again to select it. When you select a product the **Select Components** dialog box appears:

```
+-----+
| Custom - Select Products |
+-----+
| *OE Workgroup RDBMS      |
| *OE SQL                  |
| *Client Custom - Select Components |
| *4GL De                  |
| *WSA Oracle DataServer Client - Optio |
| Install OpenEdge SQL ODBC Clients |
| Change OpenEdge SQL JDBC Clients |
| Cancel OpenEdge ESQL/C Clients |
| Help *Open Client Adapter Options Basic (r) |
+-----+
| *OEBuild Utility (r) |
| *Client-Side Web Services Deploy (r) |
| Previous Menu |
| Cancel |
| Help |
+-----+
| Disk Space Required for Products: 541,597,696 bytes |
| Disk Space Required for Installation: 542,949,888 bytes |
| Disk Space Remaining After Installation: 4,294,967,295 bytes |
| Destination pathname: /usr1/bhamel/101b |
+-----+
```

The **Select Components** dialog box lists only those components that have not been previously installed. Select or deselect a component to install by highlighting the component and pressing **RETURN**. An asterisk (*) indicates that a component is selected.

3. If the selected component has subcomponents the **Select Subcomponents** dialog box appears:

```
+-----+
| Custom - Select Products |
+-----+
| *OE Workgroup RDBMS      |
| *OE SQL                  |
| *Client Custom - Select Components |
| *4GL De                  |
| *WSA Oracle DataServer Client - Optio |
| Install Ope+-----+
| Change Ope| Custom - Select Subcomponents |
| Cancel Ope+-----+
| Help *Ope| AppServer Internet Adapter (r) |
+-----+
| *OEB| *OpenEdge Adapter for SonicMQ (r) |
| *Cli| *Java Client Support (r) |
| Pre| Previous Menu |
| Can| Cancel |
| Hel| Help |
+-----+
| Disk Space Required for Products: 541,597,696 bytes |
| Disk Space Required for Installation: 542,949,888 bytes |
| Disk Space Remaining After Installation: 4,294,967,295 bytes |
| Destination pathname: /usr1/bhamel/101b |
+-----+
```

The **Select Subcomponents** dialog box lists the subcomponents for the component you selected. The symbol (r) indicates that a subcomponent is recommended and will be installed automatically unless you deselect it. Mandatory components are not displayed.

Select or deselect a subcomponent by highlighting the component and pressing **RETURN**. An asterisk (*) indicates that a subcomponent is selected.

Choose **Previous Menu** and press **RETURN** when you have selected all the subcomponents you want to add.

4. Choose **Install Selected Products** from the **Select Products** dialog box and press **RETURN**. The **Done Selecting Products Confirmation** dialog box appears:

```

+-----+
|               Done Selecting Products Confirmation               |
+-----+
| Are you sure that you are done selecting products to install? |
|               [Y=YES] [N=NO]                                   |
+-----+

```

5. Type **Y** to continue with the installation or **N** to select additional components or subcomponents. The **Copy Scripts?** dialog box appears:

```

+-----+
|               Copy Scripts?                                     |
+-----+
| Copy the scripts to /usr/bin? |
|               [Y=YES] [N=NO]                                   |
+-----+

```

OpenEdge products consist of several scripts and program modules. When you install a product, the scripts are placed in the installation directory you specify.

- Caution:** Answering **Y** might cause the OpenEdge Installation Utility to overwrite existing executables in this directory.

- Note:** If you are maintaining two versions of OpenEdge on the same machine, answer **N** to this question.

```
|-----+
|               Installing Files                |
|-----+
|               Installing subcomponent: Database Utilities (m)
|               Installing file: mkschema
|               1%█                             |
|-----+
|               [CTRL-T=Quit]                   |
|-----+
```

```
|-----|
| Tailoring OpenEdge Files |
|-----|
The OpenEdge Installation Utility is tailoring the installed OpenEdge
files.
           Please Wait.█
|-----|
```

- 5-12

Downloading executables for heterogeneous environments

The distributed architecture of OpenEdge allows you to optimize your hardware and network resources by installing components across networked machines, specifically when you are installing products such as the WebSpeed Transaction Server and the AppServer. Although some of these products' components must reside together on the same machine, you can, in some cases, distribute components to different machines, even if the machines have different hardware or operating systems. For example, you can install a WebSpeed Messenger or the NameServer on a UNIX platform and install a WebSpeed Broker and agents in Windows.

If you need either the WebSpeed Messenger executable or the NameServer executable for a platform other than UNIX, you can download the executables free of charge from <http://www.progress.com/esd>.

OpenEdge Silent installation overview

An interactive installation prompts you for input and records your values in a series of dialog boxes. The Installation Program immediately uses this data to install your OpenEdge products.

In contrast, a Silent installation is a two-step process:

- Data entered during the interactive installation process is recorded, typically in an `.ini` file. An OpenEdge installation automatically creates a `response.ini` file during the interactive installation. Although you can create your own `.ini` file, the automatically-generated `response.ini` file is a most reliable data input to perform a Silent installation.

Note: This section focuses primarily on using the `response.ini` file because this data input does not require you to perform any additional file-related tasks. Optional response file-related activities, such as editing a response file, are presented later in this section.

- The installation data captured in an `.ini` file is read programmatically to install the products through a batch, or silent, mechanism at any time. Complete and custom installation support the Silent installation feature.

The main tasks to perform a Silent installation are:

- Selecting which `.ini` file to use to capture your installation values.
- Entering the command to start the Silent installation.
- Checking the status of the installation log.

Data input options for a Silent installation

[Table 5–1](#) identifies and briefly describes the two types of data inputs you can use to perform a Silent installation.

Table 5–1: Data input options for a Silent installation

Data input options	Description
Automatically generated <code>response.ini</code> file	<p>An OpenEdge 10.1B interactive installation automatically creates a <code>response.ini</code> file that contains the installation values as you originally entered them in fields on the dialog boxes. It is stored in your <code>install</code> subdirectory in your installation directory, <i>OpenEdge-install-dir</i>. The file is immediately available for you to play back to start a Silent installation.</p> <p>See the “Understanding the Response.ini file contents” section on page 5–15 for more information and an excerpt of the <code>response.ini</code> file.</p>
User-initiated programmatic method	<p>Provides Application Partners (APs) a streamlined approach to integrate the OpenEdge installer into an application installer. Using this method, an AP can access the automatically generated <code>response.ini</code> file to programmatically create an OpenEdge installation response file; when the AP’s application is installed on a customer site, the OpenEdge installation information is read from the response file, enabling the customized install to be performed silently.</p> <p>For more information about this optional activity, see the “Creating data input option” section on page 5–21.</p>

Note: You can choose to edit the response file. However, keep in mind that any modifications to the automatically- or programmatically-generated response file can be time consuming and error prone.

Understanding the Response.ini file contents

The data captured in the automatically-generated response .ini file provides a detailed, reliable snapshot of the installation choices made during an interactive installation.

The response .ini file includes:

- A header version number and application details.
- Section labels defined by brackets for easy referencing.
- Each dialog box comment section identified with the label DESCRIPTION and the specific dialog box title.
- Easy-to-read descriptions of the fields on each dialog box.
- Only the values captured during the interactive install are stored in the response.ini file; there is no extraneous content.
- Dialog boxes that appear in the same order as presented in the online installation.
- A complete list of products installed.

The original response .ini file is initially created when you run the Silent installation; it is never overwritten. If you re-run the Silent installation to add products to an existing 10.1B installation, a new unique response .ini file is created. It is identified as response .ini .1, response .ini .2, response .ini .3, and so forth. These files will be saved to your installation directory.

Response.ini sample excerpt

The following example shows an excerpt from the automatically-generated response.ini file:

response.ini

(1 of 3)

```
; DESCRIPTION of Configuration Count
;
; ProductCount - the number of products being installed.
;

[Configuration Count]
NumberOfConfigurations=1

[Product Configuration 1]
name=progress
serial=123456789
version=10.1B1P
control=XXXXX XXXXX XXXXX
prodname=4GL Development System

;

;
; DESCRIPTION of Type and Destination
;
; path - identifies the directory in which you install your OpenEdge product
software.
; workpath - identifies the directory in which your applications, databases,
and log files will reside.
;

[Type and Destination]
type=COMPLETE
path=/usr1/bsmith/dlc101b
workpath=/usr1/bsmith/wrk101b

;
; DESCRIPTION of SonicEsbAdapter
;
; esbdomain - identifies the Sonic ESB Domain Name.
; esburl - identifies the Connection URL to the Sonic ESB.
; esbusername - identifies the User Name used to connect to the Sonic ESB.
; esbpassword - identifies the Password used to validate the User Name.
; esbpath - identifies the directory where the Sonic ESB is installed.
; esbcontainername - identifies the Sonic ESB Container Name.
;
```

response.ini

(2 of 3)

```

[SonicEsbAdapter]
esbcontainername=bespinContainer
esbdomain=Domain1
esburl=tcp://localhost:2506
esbusername=Administrator
esbpassword=Administrator
esbpath=empty

;
; DESCRIPTION of Language Default
;
; DefaultLanguage - identifies the language in which PROMSGS appears by
default.
;   -Valid values are:
;     Czech
;     Dutch
;     English - American
;     English - International
;     French
;     German
;     Italian
;     Polish
;     Portuguese
;     Portuguese - Brazilian
;     Spanish - Latin
;     Swedish
;

[Language Default]
DefaultLanguage=English - American

[Language Choice]
lang1=English - American

;
; DESCRIPTION of International Settings
;
; NOTE: For specific information please refer to the intlsets.txt file
located at the root level of the cdrom from which this information is derived.
; cpininternal - identifies the -cpininternal and -cpstream values included in
the startup.pf file.
; cpcollation - identifies the -cpcoll value included in the startup.pf file.
; cpcase - identifies the -cpcase value included in the startup.pf file.
; dateformat - identifies the -d value included in the startup.pf file.
; numsep - identifies the -numsep value included in the startup.pf file.
; numdec - identifies the -numdec value included in the startup.pf file.
;

```

response.ini

(3 of 3)

```
[International Settings]
cpinternal=ISO8859-1
cpcollation=Basic
cpcase=Basic
dateformat=dmy
numsep=39
numdec=44

[Installed Products]
ProductCount=1
Product 244=4GL Development System

[Product 244]

__Component_Oracle DataServer Client - Optio=1
__Component_SQL Database Server - Optional=1
__Component_OpenEdge SQL ODBC Clients=1
__Component_OpenEdge SQL JDBC Clients=1
__Component_OpenEdge ESQL/C Clients=1
__Component_Open Client Adapter Options (r)=1
__SubComponent_AppServer Internet Adapter (r)=1
__SubComponent_OpenEdge Adapter for SonicMQ (r)=1
__SubComponent_Java Client Support (r)=1
__SubComponent_OpenEdge Adapter for SonicESB (r)=1
__SubComponent_Web Services Admin Enabler (r)=1
__SubComponent_Web Services Schema (r)=1
__Component_Client-Side Web Services (r)=1
__SubComponent_Client-Side Security (r)=1
__SubComponent_Web Services Basic (r)=1
__SubComponent_WSDL Analyzer (r)=1
__Component_Application Debugger (r)=1
__SubComponent_Application Debugger (r)=1
__Component_OEBuild Utility (r)=1
__Component_4GL utilities (r)=1
__SubComponent_XSD-4GL (r)=1
.
.
.
```

Running the Silent installation

The command you use to initiate, or play back, the response file is the same regardless of the data input you use. The OpenEdge Silent utility runs without your intervention after you enter the command to start the process.

The syntax for running the OpenEdge Silent Installation utility in batch mode follows:

Syntax

```
proinst -b <path>/<install-ini-name> -l <path>/<logfile-name> [-n]
```

proinst

Command to initiate an OpenEdge installation.

-b<path>/<install-ini-file>

Identifies that a batch installation will be performed, and specifies the pathname and filename of the .ini file that you select to run the Silent installation. You can use the response.ini file, the install.ini file, or another .ini file that you create and name.

-l <path>/<logfile-name>

Identifies that a log file will be created, and specifies the pathname and filename of the installation log file in which the installation events will be logged.

<logfile-name>

Specifies the filename of the log file. If no filename is specified, the OpenEdge Installation Utility uses the default log filename of install.log.

If no directory is specified to which the log file is to be saved, the Installation Utility saves it to the directory identified by the first environment variable it finds among the following: \$TMP, \$TEMP, or \$TMPDIR.

-n

Indicates that the batch installation will include a progress meter, displaying details about the current installation phase and percent complete.

Example

The following syntax shows a typical Silent installation command:

```
proinst -b /test/install.ini -l /log/test.log
```

Checking the status of the Silent Installation log file

The Silent Installation process automatically generates a log file, in which all messages—error and successful installation—as reported.

The following shows an excerpt from the typical contents of the OpenEdge Installation Utility log file:

OpenEdge Installation Utility log file

```
OPENEDGE INSTALL UTILITY LOG <VERSION 10.1B> (Wed Sep 27 11:30:52 2006)
[Application]
Name=OpenEdge
Version=10.1B
Company=progress

[ResponseResult]
ResultCode=0
ResultDescription=The install completed successfully.

[CompletedEvents]
Event1=[09-27-2006 11:34:00] The Setup Utility is extracting archives

Event3=[09-27-2006 11:36:13] The Setup Utility has tailored files.
Event4=[09-27-2006 11:36:13] The Setup Utility has merged delta files.
[RuntimeStatus]
Progress=98
.
.
.
[UpdateUnixRegistry]
File=[09-27-2006 11:35:54] /etc/progress has been updated successfully.
.
.
.
[FilesTailored]
File1=[09-27-2006 11:35:54] /usr1/bsmith/dlc101b/bin/proaiw has been tailored
successfully.
File2=[09-27-2006 11:35:55] /usr1/bsmith/dlc101b/bin/proapw has been tailored
successfully.
.
.
.
[TailoringExtensions]
Extension1=[09-27-2006 11:36:31] /usr1/bsmith/dlc101b/bin/prodbgtr.dll has
been executed successfully.

[TailoringClasses]
Start=[09-27-2006 11:36:31]
Finish=[09-27-2006 11:37:40]
```

Optional data input activities

The following optional activities are also supported when you are performing a Silent installation. However, keep in mind that creating the response file manually or editing the response file is the more time-consuming and potentially error-prone approach than creating it using the automatically-generated response file method described in the [“Understanding the Response.ini file contents”](#) section on page 5–15.

Creating data input option

You can choose to record a separate response file any time you perform an interactive installation. All your installation choices are automatically recorded in a user-defined response file. If you do not specify a filename, the install creates the file `$TEMP/install.ini`.

The format and structure of any data input option is identical to that which is presented in the automatically-generated response.ini file. See the [“Response.ini sample excerpt”](#) section on page 5–16 to review an excerpt of the file’s content.

Use the following syntax to initiate a response file:

```
proinst -r [<path-to-file>\response-file]
```

`proinst`

Command to initiate an OpenEdge installation.

`-r <path-to-file>\<response-file>`

Indicates that the installation is in record mode, and specifies a pathname to and filename for the data input file to be created. If you do not provide a filename, the installation creates the filename, `install.ini`, and places it in the `$TEMP` directory.

Manually modifying data input option

You can edit any response file, whether you create it or use an automatically-generated response file. Although all sections of the response file are required, you do not need to add each of these required sections in the order presented; the installer only retrieves the specific data it needs regardless of where the information is located in the response file.

Addressing a detected JVM version

If you receive a warning message at the beginning of your installation stating that the detected JVM version does not match the version supported by OpenEdge, you can add an entry in the .ini file to allow batch installs to override this warning. Add the following entry to the [java] section of the .ini file if you want the installation to continue after detecting a mismatched JVM version:

```
jvmAllowUnsupported=yes
```

Performing postinstallation tasks

Before you run OpenEdge, you need to complete a few postinstallation tasks.

Required tasks

The following list identifies required tasks:

- **Set environment variables** — For more information on setting environment variables (including SQL), see [Chapter 8, “Working in the OpenEdge Environment on UNIX.”](#)
- **Create customized executables** — To create customized product executables, see the information on building ABL executables in [OpenEdge Deployment: Managing ABL Applications](#). Creating executables might be required for certain product configurations.
- **Convert existing databases** — After your OpenEdge installation is complete, you must convert your Progress databases to OpenEdge using the PROUTIL CONV910 utility. Note that if you have a Progress Version 8 database, you must convert it to a Version 9 database first. For instructions on converting your Progress databases to OpenEdge, see the chapter on administration utilities in [OpenEdge Data Management: Database Administration](#).

Setting AdminServer security

Once you have installed OpenEdge with products that use the AdminServer, you can optionally set user and/or group AdminServer security. You can set this option on the command line to require an individual user and/or groups of users to provide valid values during the AdminServer startup process. OpenEdge products such as the following use the AdminServer: AppServer, WebSpeed, OpenEdge Adapter for SonicMQ, the Progress Explorer, and Web Services Adapter.

The AdminServer user-group authorization feature allows you to require a level of security that enables only authenticated operating systems users and groups access to and use of the Admin Service.

To effectively set up this security option for your AdminServer use, review your security needs and current authenticated operating system users and groups to determine how you will set up this option during the OpenEdge installation process.

To implement the User-group Authorization feature on a UNIX platform, you must first successfully complete the installation program.

[Table 5–2](#) identifies and briefly describes the purpose of each command-line option.

Table 5–2: User-group parameter options

Parameter name	Syntax	Purpose
Individual user name and password required.	<code>-requireusername</code>	To require a minimum of one user ID to be resolved for each AdminServer operation before it can be executed.
Group authorization required.	<code>-admingroup group [{, :}group...]</code>	To require a minimum of one group to be resolved for each AdminServer operation before it can be executed. On a UNIX platform, a colon-separated list differentiates groups when you are specifying multiple groups on the command line.

On UNIX platforms, a group name can be any user-defined or NIS group name. UNIX can also support subgroups.

Uninstalling OpenEdge on UNIX and Linux operating systems

The `uninstall` script consolidates and formalizes the actions required to remove an OpenEdge 10.1B installation from all supported UNIX or Linux operating systems. The `uninstall` script is installed in the `install` subdirectory located within the *OpenEdge-install-dir*.

The following syntax identifies the command executed to perform the uninstall process:

```
cd OpenEdge install-dir/install/uninstall
```

Progress Software Corporation recommends using the new uninstall script to ensure the following uninstall activities occur properly:

- Uninstall third-party, embedded products such as SonicMQ®.
- Remove Release-specific entries from the `/etc/progress` file and `/etc/ProDbgCK` (Progress Debugger Check) files, and remove the install directory into which OpenEdge 10.1B was installed. In addition, the uninstall script creates an `OE<version>uninst.log` file in the home directory.

Uninstalling OpenEdge Adapter for SonicMQ and OpenEdge Adapter for SonicESB

In most circumstances, to remove the OpenEdge installation from your machine you can remove the installation directory. However, in Release 10.1B, a new procedure has been added that allows you to verify that the OpenEdge Adapter for Sonic MQ and the OpenEdge Adapter for Sonic ESB are properly uninstalled.

The following procedure describes how to uninstall either of these Sonic products. The uninstall of these Sonic products must occur **before** you remove your OpenEdge installation.

Note: This procedure only pertains to those installations that initially added either or both the OpenEdge Adapter for SonicMQ and the OpenEdge Adapter for Sonic ESB.



To perform the uninstall of these Sonic products:

1. Type the appropriate command on the command line, as shown in the following examples:

To access the SonicMQ uninstaller:

```
cd OpenEdge-install-dir/sonic/MQ6.1
```

To access the SonicESB uninstaller:

```
cd OpenEdge-install-dir/sonic/ESB6.1
```

2. Execute the following on the command line for either Sonic component:

```
uninstall.sh -silent
```

-silent

Allows you to run the uninstall in a command window.

To remove the entire OpenEdge installation, remove the OpenEdge installation directory.

Administration Utilities

This chapter provides step-by-step instructions to perform a variety of administrative tasks and details about using and managing platform-specific resources, as outlined in the following sections:

- [Using the License Update utility](#)
- [Displaying license information using the SHOWCFG utility](#)
- [Managing user licenses on all supported platforms](#)
- [OpenEdge license information](#)
- [Using OpenEdge resources in Windows platforms](#)
- [Manage memory and system configurations on UNIX platforms](#)
- [UNIX troubleshooting tips](#)
- [OpenEdge event logging](#)

Using the License Update utility

Use the License Update utility to review and, as needed, change the following license information: number of licensed users, the expiration date for an OpenEdge product, and/or update your evaluation license to an OpenEdge non-evaluation license. (Note that the License Update utility is also called the Product Update utility.)

Contact your Progress Software Corporation sales representative for a new *License Addendum* if you need to use this utility.

Changes to accommodate license updates

The License Update process streamlines the process of updating an existing product license. You can now enter a different, but valid serial number (and associated, new control numbers) to update an existing license file (`progress.cfg`). In prior releases, the update process would not accept different serial numbers; you had to uninstall an existing license and then install the newer product license.

This new update process can be used to update licenses obtained through either the product evaluation process or PSDN subscription renewal process by simply entering a new, different product serial number during the installation process to automatically update the current license data in the `progress.cfg`. Also, if want to update an evaluation license to a non-evaluation license, you no longer have to uninstall the evaluation license and then install the non-evaluation license; you can simply perform the procedure to update the License Update utility, entering your new valid serial and control codes.

In Windows-based platforms

The following procedure outlines the steps required to update your license information in a supported Windows platform.



To use the License Update utility to update your license in a Windows platform:

1. Use the Show Configuration utility (SHOWCFG) to display the product license information for each OpenEdge product installed on your system. See the [“Displaying license information using the SHOWCFG utility”](#) section on page 6–5 for instructions.
2. Choose the **License Update** icon from your OpenEdge group. After a welcome message appears, the **Serial & Control Numbers** dialog box appears.
3. Type the serial number and the new control numbers that Progress Software Corporation supplies when you upgrade your license.
4. Choose **Accept**. The **Product(s) Updated** dialog box displays the products you want to update.
5. When you are finished updating the products, choose **Done**.

On UNIX-based platforms

The following procedure outlines the steps required to update your license information on a supported UNIX or Linux platform.



To use the Product Update utility to update your license on a supported UNIX platforms:

1. Use the SHOWCFG utility to display the product configuration information stored in the OpenEdge Release 10.1B configuration file `progress.cfg`. See the [“Using the SHOWCFG utility in Windows platforms”](#) section on page 6–5 for instructions.
2. Change your current working directory to the directory where you installed OpenEdge.
3. At the system prompt, type **proupdt** and press **RETURN**.

4. When the **Welcome** dialog box appears, press **RETURN**. The **Product Configuration Data** dialog box appears:

[illegible]

5. Type your company name, the serial number, and the new control numbers Progress Software Corporation (PSC) supplies when you upgrade your license.
6. Press **ENTER**. The **Modified Product Information** dialog box appears:

```

+-----+
| Modified Product Information |
+-----+
|
| Product Name:   Query/RESULTS
|
| Old User Count: Unlimited
| New User Count: Unlimited
| Old Expiration Date: 05/05/2005
| New Expiration Date: 05/05/2008
|
| [Enter=OK]
|
+-----+

```

Press **ENTER** again to return to the **Product Configuration Data** dialog box.

7. Press **CTRL+E** to indicate that you are done.

Note: You cannot press CTRL+E from the **Serial Number** field.

Displaying license information using the SHOWCFG utility

The OpenEdge installation program prompts you to enter product information contained in your *License Addendum*. During the installation process, the installation program records the license information in the OpenEdge configuration file (`progress.cfg`). You can use the SHOWCFG utility to display licensing information, product installation, and configuration details about each OpenEdge product that you install.

Using the SHOWCFG utility in Windows platforms

The SHOWCFG utility displays product installation and configuration information for each OpenEdge product installed on your system. [Table 6–1](#) describes the different ways to run the SHOWCONFIG utility.

Table 6–1: Running the SHOWCFG utility

To run the SHOWCFG utility from the . . .	Then . . .
Start menu.	Choose OpenEdge→Config .
OpenEdge Group menu.	Double-click the Config icon.
Command line of the Proenv window.	Type the <code>showcfg</code> command.

The SHOWCFG utility opens the **OpenEdge Configuration Information** dialog box to display the product configuration information stored in the OpenEdge configuration file `progress.cfg`, which is created and modified during product installation.

Figure 6–1 shows a typical display of the **OpenEdge Configuration Information** dialog box.

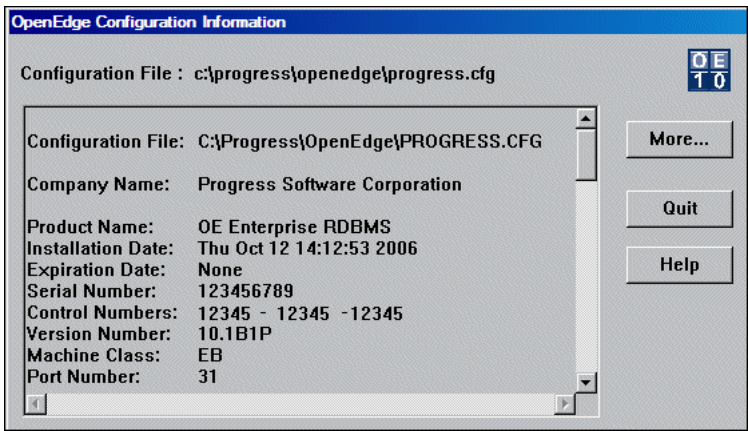


Figure 6–1: OpenEdge Configuration Information from SHOWCFG utility

Table 6–2 identifies and briefly describes the detailed information that appears for each OpenEdge product you install on your system.

Table 6–2: Display fields associated with the SHOWCFG utility

This display field . . .	Identifies the . . .
Product Name	Name of the installed product.
Installation Date	Date the product was installed.
Expiration Date	Date the license expires.
Serial Number	Number associated with the license agreement.
Control Numbers	Numbers used by the OpenEdge installation software.
Version Number	Software product version.
Machine Class	Tier code associated with the license agreement.
Port Number	Platform on which the software product is installed.

Details about SHOWCFG functions in Windows platforms

SHOWCFG searches for the `progress.cfg` file in the locations defined as PROCFG and DLC in the `progress.ini` file. To find the `progress.ini` file, SHOWCFG searches the following locations, in the order specified:

1. The current working directory.
2. The special users directory (set using the Properties option).
3. The Windows directory.

If the utility finds the `progress.cfg` file, it displays the contents in the **OpenEdge Configuration Information** dialog box.

If SHOWCFG cannot find the `progress.cfg` file, the **Open** dialog box appears so you can specify the file's location. You can also use the **Open** dialog box to specify a different OpenEdge configuration file to display. To display the **Open** dialog box, choose the **More** button in the **OpenEdge Configuration Information** dialog box.

OpenEdge does not accept optional qualifiers that point to a `.cfg` file other than `OpenEdge-install-dir:PROGRESS.CFG`.

Using the SHOWCFG utility on UNIX or Linux platforms

The SHOWCFG utility has the following syntax:

Syntax

```
<OpenEdge-install-dir>/bin/showcfg <OpenEdge-install-dir>/progress.cfg
```

For example:

```
/userdir/smith/101b/bin/showcfg /userdir/smith/101b/progress.cfg
```

The SHOWCFG utility displays the product configuration information stored in the OpenEdge Release 10.1B configuration file `progress.cfg`, which is created and modified during product installation.

Figure 6–2 shows a typical display of the product configuration.

```
Configuration File: ../progress.cfg
    Company Name: Progress Software Corporation
    Product Name: OE Workgroup RDBMS
    Installation Date: Wed Oct  4 09:17:38 2006
    Expiration Date: None
    Serial Number:  - - - -
    Control Number: - - - -
    Version Number: 10.1B
    Machine Class:  EB
    Port Number:   11
```

Figure 6–2: Product configuration details display using SHOWCFG utility

Refer to [Table 6–2](#) for an explanation of each of the display fields that appear in [Figure 6–2](#).

Displaying license information in Windows platforms

You can display product license information such as the number of users and the platforms for each OpenEdge product installed on your system.



To display your current product license information on Window platforms:

1. From the desktop, choose **Start→Run**. The **Run** dialog box appears.
2. Perform one of the following:
 - a. Type **showcfg** in the **Open** field, and choose **OK**.
 - b. At the command line, type the following command and press **ENTER**:

```
showcfg OpenEdge-install-dir\progress.cfg
```

The **OpenEdge Configuration Information** dialog box appears and displays the information you entered.

See the “[Managing user licenses on all supported platforms](#)” section on page 6–9 for more information about licensing

Managing user licenses on all supported platforms

The OpenEdge license you purchase determines how many units are allowed to run your OpenEdge products. You are responsible for making sure users comply with your license agreement. OpenEdge provides reporting capabilities to help you ensure compliance with your license agreement. This section describes:

- The OpenEdge license information that is shipped with your OpenEdge product.
- How to read the OpenEdge usage log file.
- How to produce a report of current licensed user connections.

OpenEdge license information

The *License Addendum* that accompanies your OpenEdge media package (that is, CD, DVD, or ESD download) with your OpenEdge product provides specific information about the product license you purchased, including:

- A serial number.
- A control number.
- The maximum number of units allowed by the license.

When you install OpenEdge, the installation procedure prompts you to enter product information from the *License Addendum*. The installation procedure records the license information in the OpenEdge configuration file `progress.cfg`. Use the `SHOWCFG` utility to display the product license information for each OpenEdge product installed on your system.

For more information on the `SHOWCFG` utility, see the [“Using the SHOWCFG utility in Windows platforms”](#) section on page 6–5 or the [“Using the SHOWCFG utility on UNIX or Linux platforms”](#) section on page 6–7.

Using the OpenEdge license file

OpenEdge creates a license file that records license-related information about OpenEdge database users. If the log file does not already exist, the broker creates it and places it in the same directory as the database (.db) file. The broker creates the file in the format *datasasename.lic*, where *datasasename* is the name of the database to which the user connects.

Note: If OpenEdge encounters an error while trying to open or write to the license file, the error is recorded in the database .lg file and no more entries are written to the license (.lic) file.

Reading the license file

Use a text editor to display the license file contents. The contents appear in the following order:

1. Current date.
2. Current time.
3. Number of licensed users specified by the configuration file.
4. Current number of total connections.
5. Maximum number of total connections.
6. Minimum number of total connections.
7. Current number of interactive connections.
8. Maximum number of interactive connections for the past hour.
9. Minimum number of interactive connections for the past hour.
10. Current number of batch connections.
11. Maximum number of batch connections for the past hour.
12. Minimum number of batch connections for the past hour

For example, the following sample file entry illustrates the log format:

4/26/08 9:00 25 18 23 11 17 20 11 1 5 0

When OpenEdge writes to the license file, the maximum and minimum values are reset for the next hour.

Maintaining the license file

The database or system administrator should consider archiving license files periodically. In one year, a license file accumulates 8,760 entries. These entries occupy about 440,000 bytes of disk space.

Since the license file must be closed before the administrator archives it, the administrator must first shut down the database. At that point, the license file can be either archived immediately or renamed and archived later.

Creating a usage report

To produce a report of license-related information about current OpenEdge database users, run the `licrpt.p` procedure file. The report generator input data appears:

Enter Date Range: To: Enter Start Time (hours 0 to 23):
Enter Stop Time (hours 0 to 24):
Enter time division (in hours, or 0 for complete range):
Database Name:

This is a sample output from the `l1crpt.p` procedure file:

Database Connection Counts									
Date	Period	LcnUsers	MaxTot	Excptns	MinTot	AveTot	MaxBat	MinBat	AvBat
4/11/06	8-17	100	20	0	0	10.	0	0	0.
4/13/06	8-17	100	20	0	18	19.	0	0	0.
4/16/06	8-17	100	23	0	17	20.	0	0	0.
4/20/06	8-17	100	33	0	17	25.	0	0	0.
4/24/06	8-17	100	32	0	26	29.	0	0	0.
4/26/06	8-17	100	26	0	17	22.	0	0	0.

Using OpenEdge resources in Windows platforms

OpenEdge uses several operating system resources such as shared memory and memory locks, processes, and client memory in Windows platforms. You can plan OpenEdge operations more effectively if you understand these resources.

Shared memory

Shared memory is an area in system memory that multiple users can access concurrently. OpenEdge stores shared resources in the shared-memory area, enabling multiple users and servers access to each database. OpenEdge uses semaphores and spin locks to synchronize the activities of server and self-service client processes that are connected to a database. Each process uses its semaphore or relies upon the spin lock when it must wait for a shared resource.

You can tune OpenEdge performance by reconfiguring the size of the following shared-memory buffers:

- **Database buffers** — OpenEdge reads database blocks into the database buffer pool. Larger buffers usually result in less disk I/O.
- **Before-image (BI) buffers** — OpenEdge stores BI notes in memory before writing them to disk.
- **After-image (AI) buffers** — OpenEdge stores AI notes in memory before writing them to disk.

OpenEdge also creates shared-memory tables to provide essential information on the status of each process, server, transaction, and lock. These tables enable you to control all of the database activities from one shared area.

See *OpenEdge Data Management: Database Administration* for more information about managing and improving OpenEdge performance.

Processes in Windows platforms

OpenEdge provides the following optional processes to improve performance in Windows and on UNIX platforms:

- **Asynchronous Page Writer (APW)** — The APW improves database 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 checkpointing (the process of synchronizing the buffer pool of modified blocks to the database).
- **Before-image Writer (BIW)** — The BIW improves performance by continually writing before-image buffers to disk. These writes occur in the background.
- **After-image Writer (AIW)** — The AIW improves performance by continually writing AI buffers to disk soon after OpenEdge fills the buffers.
- **OpenEdge Watchdog (PROWDOG)** — The watchdog process cleans up after improperly terminated processes by releasing locks, backing out any live transactions and releasing shared-memory locks, and disconnecting and cleaning up the server's remote clients.

See *OpenEdge Application Server: Administration* and *OpenEdge Data Management: Database Administration* for more information about managing and improving OpenEdge performance.

Manage memory and system configurations on UNIX platforms

This section describes how to manage your system's memory and configuration as outlined in the following sections:

- [Calculating memory needs.](#)
- [Managing shared memory and process resources.](#)
- [Reducing memory usage.](#)
- [Swap space.](#)
- [Shared memory and kernel configuration.](#)

Calculating memory needs

The tables in this section are provided to help you calculate the memory requirements for your system. Keep in mind that all memory usage figures are approximate and vary depending on the version of the operating system, UNIX parameters, the OpenEdge startup parameters, and the OpenEdge application you are using. For more information, see *OpenEdge Deployment: Startup Command and Parameter Reference*.

Note: The background processes APW, BIW, AIW, and PROWDG also take up memory. Remember to calculate these in your memory requirements.

Table 6–3 lists the components you use to calculate system memory requirements.

Table 6–3: Components used to calculate memory needs

(1 of 2)

Component	Symbol	Comment
Operating system	os*	Represents the memory requirements for one copy of your operating system shared in memory by all users, plus a certain percentage of physical memory to allow for operating system buffers; typically, 10%–15%.
OpenEdge	_progres*	Represents the size of one copy of OpenEdge shared in memory by all users running single-user or multi-user OpenEdge. Allow for 15%–20% deviation in the _progres value to accommodate new releases.
Database server or broker	_mprosrv*	Represents the size of one copy of the OpenEdge database broker/server shared in memory by all users running multi-user OpenEdge. Use this component only when calculating memory requirements for a system running a multi-user version of an OpenEdge product.
OpenEdge user data	proud	Represents the data area required for each user running OpenEdge. This value varies greatly, depending on the application you run and whether you use the compiler. It is also affected by many of the startup parameters. For single-user clients, the parameters are: Blocks in Database Buffers (-B), Directory Size (-D), and Stack Size (-s). For multi-user clients, the parameters are: -D, Maximum Memory (-mmax), and -s. ^{1 2}
OpenEdge server data	psd	Represents the data area required for each database server serving remote clients. (Not used for single-user or multi-user clients if the users are self-service). This space is used for communication buffers and other server memory requirements.

Table 6–3: Components used to calculate memory needs

(2 of 2)

Component	Symbol	Comment
OpenEdge broker data	pbdb	Represents the data area required by each database broker. (One database broker is required for each different database simultaneously in use in multi-user mode whether you are using remote client/servers, self-service, or both.) This value is determined by the values of startup parameters that consume memory, including: Database Buffers (-B), Lock-table Entries (-L), and Number of Users (-n). ² Note: Each increment of -n increases pbdb by 2K.

1. Use the UNIX size command to determine the exact size. See [Table 6–4](#) to determine the approximate value.
2. See [OpenEdge Deployment: Startup Command and Parameter Reference](#) for information about OpenEdge startup parameters.

[Table 6–4](#) lists the startup options that affect memory requirements.

Table 6–4: Size increments for increasing startup parameters by 1

Startup	Size increment	Affects
Blocks in database buffers (-B).	db block size (.5K, 1K, 2K, 4K, 8K)	multi-user: pbdb; single-user: proud
Directory size (-D).	100 bytes	proud
Lock-table entries (-L).	16 bytes	pbdb
Shared-memory size (-Mxs).	1K	pbdb
Number of users (-n).	2K	pbdb
Stack size (-s).	1K	proud

Table 6–5 and Table 6–6 list approximate values for each calculation component for single and multiple users running an OpenEdge installation.

Table 6–5: Single-user memory requirements

Component symbol	Memory
_progres	3MB–4MB ¹

1. This is an approximate value. Use the size command to determine the exact size. If you are using a non-OpenEdge database, your value will be larger.

Table 6–6: Multi-user memory requirements

Component symbol	Memory
_progres	3MB–4MB ¹
_mprosrv	1MB–2MB ¹

1. This is an approximate value. Use the size command to determine the exact size. if you are using a non-Open-Edge database, your value will be larger.

Table 6–7 provides the formulas to calculate the memory requirements for your system without disk swapping.

Table 6–7: Formulas for calculating memory requirements

Single-user systems	Multi-user systems
os + _progres + (number of users x proud)	os + _progres + _mprosrv + (number of databases x pbd) + (number of remote client servers x psd) + (number of users x proud)

Note: Remote client/server processes share the same code as the broker and, therefore, require no additional _mprosrv (database server or broker) memory. Each remote client/server process does require an OpenEdge server data (psd) area.

Managing shared memory and process resources

OpenEdge uses several operating system resources, such as shared memory and memory locks, processes, and client memory. You can plan OpenEdge operations more effectively if you understand these resources.

Shared memory

Shared memory is an area in system memory that multiple users can access concurrently. OpenEdge keeps resources shared by all database users in shared memory and lets multiple servers access those resources efficiently. OpenEdge uses semaphores and spin locks to synchronize the activities of server and self-service client processes that are connected to a database. Each process uses its semaphore or relies upon the spin lock when it must wait for a shared resource.

You can tune OpenEdge performance by reconfiguring the size of the following shared-memory buffers:

- **Database buffers** — OpenEdge reads database blocks into the database buffer pool. Larger buffers usually result in less disk I/O.
- **Before-image (BI) buffers** — OpenEdge stores BI notes in memory before writing them to disk.
- **After-image (AI) buffers** — OpenEdge stores AI notes in memory before writing them to disk.

OpenEdge also creates shared-memory tables to provide essential information on the status of each process, server, transaction, and lock. These tables enable you to control all of the database activities from one shared area.

See *OpenEdge Data Management: Database Administration* for more information about improving performance.

Processes on UNIX platforms

OpenEdge supports the same optional processes in Windows platforms as it does on UNIX or Linux platforms. For a list of these optional processes and a brief description of each, see the “Processes in Windows platforms” section on page 6–13.

Reducing memory usage

If you run OpenEdge and find there is not enough main memory, try the following to reduce main memory use:

- Reduce the amount of memory allocated to OpenEdge database buffers, as controlled by the `-B` startup parameter.
- Change other startup parameters, such as `-n` and `-L`.

For more information about startup parameters, see *OpenEdge Deployment: Startup Command and Parameter Reference*.

Swap space

When the amount of memory used by all processes running on a UNIX system exceeds the amount of physical memory, portions of memory are swapped to disk. A special area of the disk is reserved for this swapping. The system administrator can set the size of this area when configuring the system.

Note: Progress Software Corporation recommends that you set your swap space size to at least twice the size of your system memory.

A UNIX system can deadlock while accessing the disk when the swap space is used up. This can happen when too many large processes are running simultaneously. If you expect to have a larger than normal number of users, or if OpenEdge memory requirements are larger than your typical process, consider increasing the amount of swap space available on your system. Before you change the size of the swap area, back up and reformat the disk.

The UNIX user set-ID bit is turned on for the OpenEdge program module. Consequently, even though there might be no active OpenEdge users, this module remains in the UNIX swap area on disk until you shut down the system.

Shared memory and kernel configuration

In OpenEdge, the multi-threaded architecture makes heavy use of file descriptors, shared memory, and semaphores. Allocation of these resources is controlled by system configuration parameters. On most systems, these parameters are set to values appropriate for OpenEdge applications. However, in some cases, one or more parameters might not be set optimally, thereby limiting the number of OpenEdge users. If you have to reset the parameters, you must reconfigure your kernel. See your operating system documentation for information on reconfiguring your operating system kernel.

The optimal parameter settings depend on the system, the application, the number of users, and some minor factors. [Table 6–8](#) lists the crucial parameters and provides guidelines for choosing adequate values for each one.

Table 6–8: Shared memory and semaphore parameter settings *(1 of 2)*

Parameter	Meaning	Optimal setting
SHMMNI	Maximum number of shared memory (SHM) identifiers.	Current value or system default + (Total OpenEdge memory requirement)/SHMMAX.
SHMSEG	Maximum number of SHM segments a single process can attach.	4–8.
SHMALL	Maximum number of in-use SHM segments.	System default; increase if many databases are active simultaneously; decreasing -B, -n, and -L startup parameters decreases SHM requirements.
SHMMAX	Maximum SHM segment size.	System default; increase if you get OpenEdge error 1135.
SEMMNI	Number of semaphore (SEM) IDs; each represents an array of SEMs.	1 per active multi-user database.
SEMMSL	Maximum number of semaphores per SEM ID.	(Max-local-users-on-any-database + Max-#servers-on-any-database + 4).
SEMMNS	Total semaphores in the system.	(SEMMSL x #active-databases).

Table 6–8: Shared memory and semaphore parameter settings (2 of 2)

Parameter	Meaning	Optimal setting
SEMMNU	Number of semaphore undo structures.	Same value as SEMMNS.
MAXUMEM	Maximum address space for a single user.	$\geq \text{server size process} + \text{SHMSEG} * \text{SHMMAX}$.

The parameter settings in [Table 6–8](#) are guidelines. Parameter values near these are acceptable in most cases, but a particular system or application might require increasing the limits.

If shared memory or semaphores are allocated incorrectly, OpenEdge displays an error message when it attempts to start an additional user or server. For example, if SEMMNS is set too low, PROSERVE fails and displays the following message:

```
Server: Semaphore limit exceeded
Server: **The server terminated with exit code (X) (800)
```

Change the relevant parameter values and reconfigure the kernel in response to semaphore or shared-memory errors at startup. [Table 6–9](#) lists the parameters that you might have to raise in response to various OpenEdge error codes.

Table 6–9: Error codes and kernel reconfiguration parameters (1 of 2)

Error code	Parameter to increase
1081	SEMMNU
1093	SEMMSL or SEMMNS
1130	SEMMSL
1131	SEMMNI and SEMMNS
1135	SHMMAX, MAXUMEM, and MAXUP

Table 6–9: Error codes and kernel reconfiguration parameters (2 of 2)

Error code	Parameter to increase
1137	SHMMNI
1175	SHMSEG, MAXUMEM, and MAXUP
1195	SEMMNS

Note: The Blocks in Database Buffers (-B), Lock-table Entries (-L), and Number of Users (-n) startup parameters all affect shared-memory usage. The Number of Users (-n) and Maximum Servers (-Mn) parameters affect semaphore usage (each user or server process uses one semaphore). Before reconfiguring your kernel to increase shared memory or semaphore allocation, see whether you can lower these startup values.

UNIX troubleshooting tips

This section provides issues to consider when troubleshooting an installation as described in the following:

- [Error messages.](#)
- [Altered or missing progress.cfg file.](#)
- [Tailoring startup scripts.](#)

Error messages

Table 6–10 lists some typical error messages, their probable causes, and where to find their solutions.

Table 6–10: Error messages

Error message	Cause	Solution
Unable to read progress.cfg, reason -1. (1732)	The progress.cfg file is altered or missing.	See the “ Altered or missing progress.cfg file ” section on page 6–23.
Module-name not found.	The environment variables are not set correctly or not installed.	See the “ Tailoring startup scripts ” section on page 6–24.
Error 304 and 305.	The ULIMIT is set too low.	Reset your ulimit.

Altered or missing progress.cfg file

If you receive the following error message, the progress.cfg file has been altered or deleted from the directory where you installed your OpenEdge products:

```
Unable to read progress.cfg, reason=-1.
```

Reinstall the OpenEdge product.

Caution: Do not alter or delete the progress.cfg file, or the OpenEdge broker startup will fail.

Table 6–11 lists the reasons for an altered or missing progress.cfg file.

Table 6–11: Reasons for altered or missing progress.cfg file (1 of 2)

Reason	Description
-1	Could not find <i>OpenEdge-install-dir/progress.cfg</i> .
-4	Bad checksum. Invalid file.

Table 6–11: Reasons for altered or missing progress.cfg file (2 of 2)

Reason	Description
-6	Could not read the specified number of bytes; the file is truncated.
-7	Could not allocate enough memory to read the configuration file.

Tailoring startup scripts

Typically, the installation procedure automatically tailors the startup scripts for the OpenEdge products you install. Tailoring involves setting each script’s environment variable to point to the directory where you installed the product referenced by the script. If the installation procedure is interrupted before the script tailoring is complete or if the normal installation procedure is not used, you might have to tailor the scripts manually.

Depending on the products you purchase and install, your OpenEdge installation provides the required scripts. Some of the OpenEdge startup scripts are shown in the following table:

adaptman	mpro	proadsv	prooidrv
aiaman	mssman	proaiw	proserve
asbman	nsman	proapw	proshut
bpro	odbman	probiw	prowdog
dbman	oraman	probrkr	wsaman
mbpro	pro	prooibrk	wtbman

Note: The scripts listed above are located in *OpenEdge-install-dir/bin*.

If the automatic tailoring did not take place, you receive the following error message when you try to start your OpenEdge product:

`module-name not found`

The *module-name* is the OpenEdge module that the script is trying to start. For example, if the script is *pro*, the module name is *_progres*.

**To tailor your startup scripts manually:**

1. Use any text editor to edit the scripts.
2. Look for the following syntax:

```
env_variable=${env_variable-pathname}; export env_variable
```

3. Change the pathname to the full pathname of the directory where you installed your OpenEdge product. For example:

```
DLC=${DLC-/usr/grp/dlc};export DLC
```

OpenEdge event logging

OpenEdge logs significant database events such as OpenEdge startup parameter settings, startup error messages, shutdown messages, system error messages, and application-related events, as described in the following sections:

- [OpenEdge event log file.](#)
- [Managing the OpenEdge event log file size.](#)
- [Event logging in Windows.](#)

OpenEdge event log file

The OpenEdge event log is a text file that contains a history of significant database events, such as OpenEdge startup parameter settings and startup, shutdown, and system error messages. This file has a .lg extension.

Managing the OpenEdge event log file size

The event log (LG) file expands as you use the database. If it becomes too large, you can use the PROLOG utility, in either an offline or online mode, to reduce the event log file's size. Using the PROLOG utility, you can:

- **Truncate log entries offline** — To remove old log entries. To remove log entries from an LG file, use the OpenEdge Log Maintenance (PROLOG) utility or a text editor. The syntax to use the PROLOG utility in the offline mode is presented in the [“Remove old log entries”](#) section on page 6–26.
- **Truncate log file entries online** — To remove entries in the database log file while the database is online. The online activity is intended to help you avoid bringing the database down and restarting it after the database has been truncated. Using this approach, the need to shutdown the database to archive the log file is eliminated. However, keep in mind that it is possible to lose some messages while performing this procedure due to the nature of the real-time processing. The syntax to use the PROLOG utility in the online mode is presented in the [“Truncate the database log file”](#) section on page 6–27.

Caution: During the time in which the multi-stepped online truncation process occurs, some messages written to the log file might be lost because the database is neither quiet nor latched/locked to prevent writes.

Remove old log entries

This section describes topics related to removing and truncating log file entries.



To remove old log entries from an event log file, enter the following command:

```
prolog database-name
```

The PROLOG utility removes all but the most recent entries from the log file. For more information, see the details about PROLOG in the [“Truncate the database log file”](#) section on page 6–27 and in *OpenEdge Data Management: Database Administration*.

Truncate the database log file

The purpose of this activity is to allow you to truncate a database log file that exists and the file's size is greater than 3072 bytes.

Truncating a log file online

The online activity is intended to help you avoid bringing the database down and restarting it after the database has been truncated; the need to shutdown the database to archive the log file is eliminated. An online truncation log file records the start and end of truncation activities and records errors to indicate when a truncation failed, as shown:

```
prolog database-name [-online]
```

`prolog`

Enables truncation of a log file. By default, a log file is truncated offline.

database-name

The name of the database to be truncated.

`-online`

Using the option `-online`, you do not have to shutdown and restart the database to truncate the database log file.

The online truncation option copies the last 3072+ bytes to a buffer, truncates the file, and then copies the buffer to the file.

Note: Keep in mind that if the `-online` option is used, the `prolog` command can truncate a log file even if the database is in use.

For more information about the syntax associated with these online and offline activities, see [OpenEdge Data Management: Database Administration](#).

Event logging in Windows

In addition to the OpenEdge event log, the OpenEdge Server writes events to the Event Log. The Event Log is the object that enables Windows users to view the status of applications, security, and system processes, and to view their associated events. OpenEdge is an application process and, as such, it writes Progress events to the Application Event Log. You use the Event Viewer to see the Event Log's contents. You can customize the Event Viewer so that it displays only the event types that you want to view. You access the Event Viewer through the Administrative Tools program group.

[Table 6–12](#) describes the components that enable the OpenEdge service to log messages to the Application event log database.

Table 6–12: Progress event logging components

Component	Function
Event viewer	The standard front-end that enables users to view the Event Log.
Event log	The standard Windows database that records event information.
CATEGORY.DLL	The OpenEdge resource that contains the 14 categories into which OpenEdge messages might fall.
PROMSGS file	The OpenEdge object that contains a single language version of the OpenEdge messages. OpenEdge supplies a PROMSGS file for each supported language version of Progress. The PROMSGS file is installed to the <i>OpenEdge-install-dir</i> location. See Appendix D, “OpenEdge National Language Support,” and OpenEdge Development: Internationalizing Applications for more information on the PROMSGS file.

Managing OpenEdge events in Windows

You can define the level of event logging that you want your OpenEdge application to support by using either the Event Level Environment Variable (EVTLEVEL) or the Event Level startup parameter (`-evtlevel`).

Table 6–13 describes the valid Event Level values.

Table 6–13: Event level values

Value	Description
None	No OpenEdge events are written to the event log.
Brief	OpenEdge messages defined as Error and Warning messages are written to the event log.
Normal	OpenEdge messages defined as Error and Warning messages are written to the event log. In addition, any Progress message that is normally written to the log file (.lg) is also written to the Event Log. This is the default.
Full	Every message generated by OpenEdge is written to the Event Log. Any OpenEdge messages generated using the Message Statements are also written to the log file.

Understanding the Windows application event log components

The components of the Windows Application Event Log are standards defined by Windows. Figure 6–3 illustrates the Windows Application Event Log components when shown through the Event Viewer.

Type	Date	Time	Source	Category	Event	User	Comp...
Error	10/3/2006	5:23:44 PM	Userenv	None	1085	SYSTEM	NBBH...
Warning	10/3/2006	5:23:44 PM	SceCli	None	1202	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	2690	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	6490	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	6489	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	0	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	2689	N/A	NBBH...
Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	5067	N/A	NBBH...
Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	12699	N/A	NBBH...
Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	13547	N/A	NBBH...
Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	7161	N/A	NBBH...
Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	5326	N/A	NBBH...
Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	451	N/A	NBBH...
Error	10/3/2006	3:24:39 PM	Userenv	None	1085	SYSTEM	NBBH...
Warning	10/3/2006	3:24:39 PM	SceCli	None	1202	N/A	NBBH...
Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	12699	N/A	NBBH...
Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	13547	N/A	NBBH...
Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	7161	N/A	NBBH...
Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	5326	N/A	NBBH...
Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	451	N/A	NBBH...
Error	10/3/2006	1:49:34 PM	Userenv	None	1085	SYSTEM	NBBH...
Warning	10/3/2006	1:49:34 PM	SceCli	None	1202	N/A	NBBH...
Information	10/3/2006	1:01:56 PM	SPORTS2_PROGRESS	PROGRE...	334	N/A	NBBH...
Information	10/3/2006	1:01:56 PM	SPORTSORA_PROGR...	PROGRE...	334	N/A	NBBH...
Information	10/3/2006	1:01:56 PM	TESTORACLE1_PRO...	PROGRE...	334	N/A	NBBH...
Information	10/3/2006	1:01:56 PM	TESTSCHEMA_PROG...	PROGRE...	334	N/A	NBBH...
Information	10/3/2006	12:57:50 PM	ORACLE_TEST_PRO...	PROGRE...	334	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	2690	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	6490	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	6489	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	0	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	2689	N/A	NBBH...
Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	5067	N/A	NBBH...

Figure 6–3: Windows application event log components

Table 6–14 describes how Progress uses the Application Event Log columns.

Table 6–14: Windows application event log components

Log component	Log information
Type	Identifies the type of message such as Information , Warning , or Error .
Date	Identifies the date the event occurred.
Time	Identifies the time the event occurred.
Source	<p>Source of the event. This is the name of the connected Progress database, if a database is connected. If no database is connected, then “Progress” is listed.</p> <p>If you are using the Progress AppServer, “Progress” is also the default source for Progress AppServer messages; however, you can override the default source name by specifying the <code>-logname</code> AppServer broker startup parameter.</p>
Category	<p>Provides information to help you isolate the cause of the message displayed in the Event Log. Progress supports 14 event categories. The event categories are: AIW, APW, BACKUP, BIW, DATASERVER, MON, OIBRKR, OIDRVR, Progress, RFUTIL, SERVER, SHUT, USER, and PROWDG. When no database is connected, Progress is specified as the category.</p> <p>All categories reside in a file called <code>category.d11</code>. These categories correspond to the existing categories of events that are displayed in the <code>progress.lg</code> file (AppServer broker and application server events are displayed in the AppServer log file, <code>proapsv.lg</code>). (Note that DATASERVER is not included as a category in the standard <code>progress.lg</code> file.)</p>
Event	Associates to the Progress message that was generated. These are the same message numbers that are displayed in the standard database <code>.lg</code> file.
User	Identifies the user logged in to the Windows workstation where the event occurred.
Computer	Identifies the name of the Windows workstation where the event occurred. The Event Viewer enables you to get more information about events by double-clicking on any event.

You can view additional information about an event by double-clicking on it. Windows displays the **Event Properties** screen, as shown in [Figure 6-4](#).

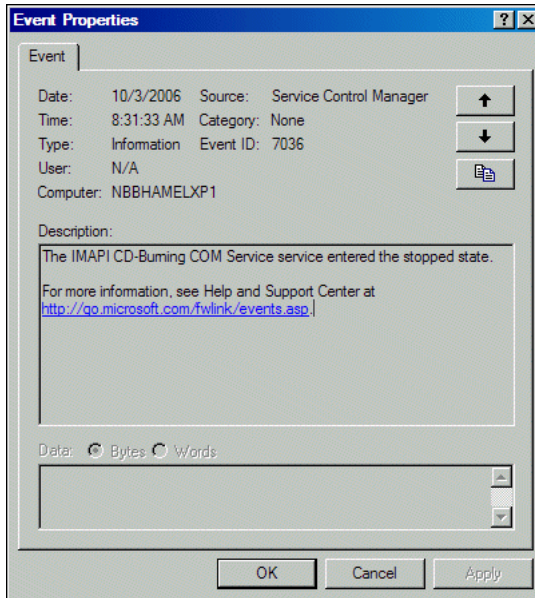


Figure 6-4: Windows application Event Properties dialog box

The **Event** tab displays details about the event you initially select. You can also use the arrow controls on the **Event** tab to scroll through detailed information about the other events that appear on the Windows application event log components viewer as shown in [Figure 6-3](#).

Windows Event log and registry

Windows requires applications that use the Event Log to be bound to all of the necessary components. For Progress this means that the PROMSGS.DLL and the CATEGORY.DLL must be bound to any Progress database. Progress stores this information in the registry. Progress makes the registry entries and performs any binding operations that are necessary when you initially access a database. When Progress binds the DLL files to the database, it writes the fully qualified pathname to the registry. If you delete the database, you must manually remove the associated data from the registry. If you move the location of the DLLs after you access the database, you must manually edit the registry data.

The Progress components can be found in the following location in the registry:

```
HKEY_LOCAL_MACHINE
SYSTEM
  CurrentControlSet
    Services
      EventLog
      Security
      System
      Application
      PROGRESS
        <Database Name>
```

See the Microsoft documentation for more information about editing registry files.

When OpenEdge tries to find the DLLs before this information is included in the registry, it performs the search according to the sequence of the following rules:

1. OpenEdge searches the current directory.
2. If the DLL is not in the current directory, OpenEdge searches the directory where the Progress executable is located.
3. If the DLL is not in the same directory as the OpenEdge executable, OpenEdge searches the user's path.

If the DLL is not in the user's path, OpenEdge generates a message stating that the DLL cannot be found, and it writes a message to the OpenEdge log file.

Part II

Configuration

Chapter 7, Working in the OpenEdge Environment in Windows

Chapter 8, Working in the OpenEdge Environment on UNIX

Chapter 9, Managing OpenEdge Key and Certificate Stores

Chapter 10, Configuration

Chapter 11, Starting and Running OpenEdge

Working in the OpenEdge Environment in Windows

This chapter orients you to how your OpenEdge environment works in Windows, reviewing and highlighting some key features. This chapter also provides steps to maintain OpenEdge versions on your system, as described in the following sections:

- [Reviewing environment variables](#)
- [Windows registry and the progress.ini file](#)
- [Setting OpenEdge Program Item properties](#)
- [Using the Proenv utility](#)
- [Getting started with the AdminServer](#)
- [OpenEdge products supported by the AdminServer](#)
- [Creating an OpenEdge database](#)

- [Configuring an OpenEdge Database server](#)
- [Running OpenEdge](#)
- [Maintaining OpenEdge and Progress](#)
- [OpenEdge key and certificate stores](#)

Reviewing environment variables

By default, the OpenEdge installation program tailors all the necessary OpenEdge and Java environment variables to the directories where they are installed. For example, the installation automatically sets the %DLC% environment variable to your OpenEdge installation path.

This section briefly reviews some system and Java environment variable details of which you should be aware. Also, [Table 7–1](#) in the “[Windows registry and the progress.ini file](#)” section on page 7–4 lists additional supported environment variables.

System environment variables

The %DLC% environment variable is not set at the system level and should not be changed. After installing OpenEdge, however, you can set environment variables to suit your own preferences. You can use Proenv to set the %DLC% environment variable to the directory where OpenEdge is installed.

Caution: Although editing environment variables is an option, this procedure is not recommended if more than one version of an OpenEdge product exists on the same system.

For more information on environment variables, see the information on maintaining user environments in [OpenEdge Deployment: Managing ABL Applications](#), or see your specific product documentation.

Latest information updates

Before you continue, consult *OpenEdge Release Notes*. These notes contain the latest information about the current release that the OpenEdge documentation set might not include. Progress Software Corporation ships release notes in Microsoft Write (readme.wri) format. Click the **Release Notes** icon in your OpenEdge program group, or access `Readme.pro` with any text editor.

Java environment variables

OpenEdge bundles the Java Runtime Environment (JRE) component and the Java Development Kit (JDK) component with certain products that you install. For more information, see the “[Java considerations](#)” section on page 1–2.

JDKHOME

Java is used by some products, such as WebSpeed, the AppServer, and SQL, for product functionality. After you install any of these products, you should verify that the JDKHOME value is set correctly in the registry. The value must be set to the directory where the JDK included in the OpenEdge installation resides (for example, C:\Progress\OpenEdge\jdk).

You can verify the JDKHOME value in the following location in the registry:

`HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\version10.1B\JAVA`

Windows registry and the progress.ini file

Applications running in Windows 2000, Windows 2003, or Windows XP Professional rely on the registry for startup information, such as color, font, and key bindings. Variables presented in this section are reserved for use by the OpenEdge installation. The installation variables have already been defined.

Caution: You should proceed with extreme caution if you are considering a change to any of the variables listed in the following sections.

Information from the `progress.ini` resides under the following registry keys:

`HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\version10.1B`

`HEKY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\version10.1B`

Note: If you modify the `progress.ini` information, you must run the `ini2reg` utility. This utility updates the information in the registry.

See [OpenEdge Deployment: Managing ABL Applications](#) for more information on the `progress.ini` file and the `ini2reg` utility.

Environment variables

OpenEdge supports some environment variables for graphical user interface (GUI) clients in the [Startup] section of the `progress.ini` file. OpenEdge supports environment variables for character clients, such as the AppServer and WebSpeed Agents, in the [WinChar Startup] section of the `progress.ini` file.

Table 7-1 lists the supported environment variables.

Table 7-1: Supported environment variables (1 of 5)

Variable	Progress.ini file section	Description	Example
DLC ¹	[Startup] [WinChar Startup]	The absolute pathname of the directory where you installed your OpenEdge system software. By default, the installation utility sets this variable.	set DLC=C:\Progress\ OpenEdge
EVTLEVEL	—	Windows 2000 only. Specifies the level of information that OpenEdge writes to the 2000 Application Event Log. You can specify one of the following cases: <ul style="list-style-type: none">• None — No OpenEdge events are written to the Event Log.• Brief — OpenEdge Error and Warning messages are written to the Event Log.• Normal — OpenEdge Error and Warning messages are written to the Event Log along with any OpenEdge messages that are normally written to the log file (.lg). This is the default.• Full — OpenEdge Error, Warning, and Informational messages are written to the Event Log along with any messages generated by the Message Statement.	set EVTLEVEL = NORMAL

Table 7–1: Supported environment variables*(2 of 5)*

Variable	Progress.ini file section	Description	Example
PATH	—	<p>A list of directory paths separated by semicolons. When you run a program or batch file, the system searches for it in the current directory. Then it searches in the directory paths defined in PATH in the order they are mentioned.</p> <p>Your PATH should include any directory pathname that contains a program or batch file you want to run. Also, each directory pathname should include the drive letter of the disk that contains the directory. PATH is a system environment variable, not an OpenEdge environment variable. Set it in a manner appropriate for the operating system instead of in the registry or in progress.ini.</p> <p>Different OpenEdge products require different PATH settings. To set up PATH for your OpenEdge product, follow the instructions provided in the “Reviewing the Windows platform installation directory structure” section on page 3–17.</p>	set PATH=%PATH%;%DLC%\BIN;%DLC%
OEBUILD	[Startup]	The pathname of the directory that contains items referenced in link scripts produced by the OEBUILD utility. By default, the installation utility sets this variable.	set OEBUILD=C:\Progress\OpenEdge\OEBUILD

Table 7–1: Supported environment variables*(3 of 5)*

Variable	Progress.ini file section	Description	Example
PROCFG	[Startup] [WinChar Startup]	The filename (or full pathname) of your product's configuration file. The configuration file is a data file that identifies the OpenEdge products and components that you are licensed to use. Reset PROCFG if you have moved your configuration file from the directory where you installed OpenEdge.	set PROCFG=%DLC%\PROGRESS.CFG
PROCONV	—	The filename (or full pathname) of the OpenEdge convmap.cp file. The convmap.cp file is a binary file that contains all of the conversion tables that are available to OpenEdge. See OpenEdge Development: Internationalizing Applications for more information on the convmap.cp file.	set PROCONV=%DLC%\CONVMAP.CP
PROMSGS	[Startup] [WinChar Setup]	The full pathname of your OpenEdge error messages file. The default value is %DLC%\promsgs. Set the PROMSGS environment variable only if you want to use an error messages file different from the default PROMSGS file in the %DLC% directory.	set PROMSGS=C:\Progress\OpenEdge\PROLANG\GER\PROMSGS.GER
PROPATH	[Startup] [WinChar Setup]	A list of directory paths separated by commas. By default, the installation utility sets this variable.	set PROPATH=., C:\Progress\OpenEdge

Table 7–1: Supported environment variables*(4 of 5)*

Variable	Progress.ini file section	Description	Example
PROSTARTUP	—	The pathname of the OpenEdge default startup parameter file, <code>startup.pf</code> . This file is read by all OpenEdge modules at startup; it must exist for OpenEdge to execute properly.	<code>set PROSTARTUP=C:%DLC%\STARTUP.PF</code>
JDKHOME	—	Establishes the top-level directory for the Java Developer's Kit (JDK).	<code>set JDKHOME=%DLC%\jdk</code>
JREHOME	—	Establishes the top-level directory for the Java Runtime Environment (JRE).	<code>set JREHOME=%DLC%\jre</code>
JFCHOME	—	Establishes the top-level directory for the Java Foundation Classes (JFC).	<code>set JFCHOME=%DLC%\jfc</code>
JDKCP	—	Sets the classpath for <code>class.zip</code> ; Java Developer's Kit (JDK) only.	<code>set JDKCP=%variable-name%/lib/class.zip</code>
JRECP	—	Sets the classpath for Java Runtime Environment (JRE); if no JRE, then it sets classpath for JDK.	<code>set JRECP=%variable-name%/lib/rt.jar</code>
JFCCP	—	Sets the classpath for Java Foundation Classes (JFC) only.	<code>set JFCCP=%variable-name%/swingall.jar</code>
PROGRESSCP	—	Contains a list of paths, jar files, and zip files for running Progress-specific products.	<code>set PROGRESSCP=%variable-name%/java/progress.zip</code>
CLASSPATH	—	OpenEdge correctly sets the appropriate classpath variable based on the platform in use.	<code>set CLASSPATH=\$JDKCP;\$JFCCP;\$PROGRESSCP</code>

Table 7–1: Supported environment variables (5 of 5)

Variable	Progress.ini file section	Description	Example
JIT	—	Sets the just-in-time compiler correctly.	set JIT="-nojit"
JVMEXE	—	Sets the Java virtual machine to run correctly.	set JVMEXE=jre

1. The DLC variable is set in the various command scripts and in the registry; the variable is not set at the system level.

Additional details for Java-related environment variables

The `JavaTools.properties` file is a common text file that contains configuration information for all ABL clients. The `JavaTools.properties` file is located in *OpenEdge-instal-dir/properties/* directory. The configuration information and settings defined in the `JavaTools.properties` file offer can provide more information to you than the Java-related environment variables.

The `AdminServer plugins.properties` file, a common text file that contains configuration details for all OpenEdge databases, is another valuable resource with which you should be familiar. It contains information for plugins that can be loaded and managed by the AdminServer. The `AdminServer plugins.properties` file is also located in `$DLC/properties/` directory. For more information about these files, see [Chapter 10, “Configuration.”](#)

Caution: Do not make user-modifications to the `JavaTools.properties` files as these properties support OpenEdge and Progress products only. Contact Progress Technical Support if you want to modify these properties.

Setting OpenEdge Program Item properties

Although the OpenEdge Installation Utility creates an OpenEdge Group with Program Items, you must set item properties such as startup parameters and buffer pools.

For information on OpenEdge startup parameters, buffer pools, and related topics, see *OpenEdge Deployment: Startup Command and Parameter Reference*, the OpenEdge DataServer Guides (*OpenEdge Data Management: DataServer for Microsoft SQL Server*, *OpenEdge Data Management: DataServer for ODBC*, and *OpenEdge Data Management: DataServer for ORACLE*) and *OpenEdge Deployment: Managing ABL Applications*.

For example, to change the properties for an OpenEdge Program Item in Windows 2000, highlight the item and choose **File→Properties** in Windows Explorer. See the appropriate Windows documentation for more information about setting Program Item properties.

Using the Proenv utility

The %DLC% environment variable is not set at the system level. The Proenv utility can automatically set the %DLC% environment variable to the directory where OpenEdge is installed. It then adds %DLC%/bin to your PATH.

To start the Proenv utility from the desktop, choose **Start→Programs→OpenEdge** (or the actual directory where you installed **OpenEdge→Proenv**). You can also start it from the command line. Proenv opens a DOS window, sets the environment variables, and then changes the current directory to the working directory you set when you installed OpenEdge.

Getting started with the AdminServer

The AdminServer is the central controlling element of the Progress Explorer Framework. It facilitates the tasks associated with managing and configuring your installation.

This section briefly introduces the AdminServer. Details about the AdminServer, its role in the the Progress Explorer Framework, and the tasks to use the AdminServer, see [Chapter 10, “Configuration.”](#)

OpenEdge products supported by the AdminServer

An AdminServer is installed on every system where you install any of the following OpenEdge products:

- OpenEdge databases: OpenEdge Personal RDBMS, OpenEdge Workgroup RDBMS, and OpenEdge Enterprise RDBMS.
- DataServers: OpenEdge DataServer for ORACLE and OpenEdge DataSever for MS SQL Server, and DataServer for ODBC.
- OpenEdge Application Server—Basic.
- OpenEdge Application Server—Enterprise.
- OpenEdge Studio.
- OpenEdge Architect.
- Web Services Adapter.
- AppServer Internet Adapter (AIA)
- OpenEdge Adapter for SonicMQ.
- OpenEdge Adapter for Sonic ESB.
- NameServer.
- OpenEdge Development Server
- OpenEdge NameServer Load Balancer.
- WebSpeed Workshop.
- OpenEdge Replication.
- OpenEdge Replication Plus.

In Windows-based systems, the AdminServer starts automatically and runs as a service. The AdminServer must be running to use any of the Progress Explorer configuration tools, or any of the following command-line managing or validating utilities. For details about these topics, see the [“Overview of the Progress Explorer Framework”](#) section on page 10–2.

AdminServer considerations

Note the following points that pertain to AdminServer usage:

- Before you start a WebSpeed or AppServer application, you must start the AdminServer.
- The AdminServer User-Group Authorization feature requires that you have privileges set to allow you access and operational privileges for the AdminServer. See the Installation Utility online help for detailed procedures on how to set up this feature.
- The AdminServer must be running before you can use the OpenEdge Explorer to configure and manage your applications.
- The DLC directory for a remotely-enabled AdminServer cannot contain spaces.

AdminServer group name conventions and restrictions

During or after the installation process, you can optionally establish AdminServer authorization options for OpenEdge products that support the AdminServer. These options are:

- **User Authorization** — To require each individual user to provide a valid user name and password before the AdminServer can be started.
- **Group Authorization** — To setup user-defined group names for which operational privileges, at a group level, are required. Group name definitions must conform to specific guidelines.

The procedures to establish AdminServer authorization options are located in the Windows online help system under these topic titles: “Establishing AdminServer Authorization Options during the Installation” and “Selecting the Authorization Feature when Starting the AdminServer.”

Additional AdminServer-related details are presented in [Appendix G, “AdminServer Authorization and Authentication.”](#)

Creating an OpenEdge database

When you create an OpenEdge database, you can either create a new database or convert an existing database:

- Use the PRODB command, the Data Administration Tool, or the Data Dictionary to create a new OpenEdge database.
- Convert an existing OpenEdge or Progress database to OpenEdge.

For more information on creating databases, see [OpenEdge Data Management: Database Administration](#).

Configuring an OpenEdge Database server

You can also create and configure an OpenEdge database server.



To use OpenEdge features with a server:

1. In Windows operating systems, from the desktop, choose **Start→All Programs→Control Panel→Administrative Tools→Services**. Verify that the status for the **AdminService for OpenEdge Release 10.1B** is **Started**.

Note: If **Administrative Tools** is not available, right click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** check box, then choose **OK**.

2. Use the Progress Explorer Tool to add a database configuration. (You cannot create the physical database with the Progress Explorer Tool.)

For more information on the AdminServer and the Progress Explorer, see the [“Understanding and Using the AdminServer”](#) section on page 10–16 and the Progress Explorer online help.

Running OpenEdge

Select an icon from the OpenEdge Program Group to begin running your applications. Note that WebSpeed products might need additional setup requirements.

Caution: Never run OpenEdge products from the directories in which you installed them. Doing so could result in changes to the software that affect its proper operation.

For complete information about starting OpenEdge or WebSpeed products, see either *OpenEdge Getting Started: Progress OpenEdge Studio* or *OpenEdge Getting Started: WebSpeed Essentials*.

Note: Before you start an OpenEdge or an Application Server application, you must start the AdminServer. The AdminServer must be running before you can use the Progress Explorer to configure and manage your applications. For details, see the “[Understanding and Using the AdminServer](#)” section on page 10–16.

Maintaining OpenEdge and Progress

To maintain OpenEdge along with one or more versions of Progress or OpenEdge on your system, you perform a typical installation (that is, a complete or custom installation as described [Chapter 3, “OpenEdge Installation Prerequisites”](#)), noting these exceptions:

- When you are prompted for a **Destination Directory**, make sure the directory you specify is not the same as for other installed versions. Type the pathname of a separate directory in which to install OpenEdge.
- Redefine your PATH, using the Proenv command-line utility.
- To run AdminServers for OpenEdge and Progress, you must set unique -port and -adminport as described in the “[Understanding and Using the AdminServer](#)” section on page 10–16.

Note: To access previous version tools or utilities, you must use complete pathnames.

OpenEdge key and certificate stores

All OpenEdge server and client components that implement Secure HTTP (HTTPS) or Secure Socket Layer (SSL) connections require access to private keys and digital certificates to negotiate these connections and to enable them to function securely.

For all OpenEdge components, OpenEdge provides utilities that allow you to install and manage keys and digital certificates (in key stores and certificate stores) so the components can use them. For Open Clients and Web services clients, OpenEdge provides utilities for some clients or relies on utilities provided by the client platform to manage the required certificate stores.

For more information on managing certificate stores for Open Clients and Web service clients, see *OpenEdge Development: Open Client Introduction and Programming*. For details about using the OpenEdge utilities to manage key stores for OpenEdge servers and manage certificate stores for OpenEdge clients, see [Chapter 9, “Managing OpenEdge Key and Certificate Stores”](#) and [Appendix C, “Command and Utility Reference.”](#)

Working in the OpenEdge Environment on UNIX

This chapter explains how to run OpenEdge on UNIX, as described in the following sections:

- [Default environment variables settings](#)
- [UNIX environment variables](#)
- [Setting Java environment variables](#)
- [Setting SQL client environment variables](#)
- [Using the Proenv utility](#)
- [Getting started with the AdminServer](#)
- [Understanding the built-in terminal definitions](#)
- [Terminal issues](#)
- [Terminal identifiers](#)
- [OpenEdge key and certificate stores](#)

Default environment variables settings

By default, the OpenEdge installation program tailors all the necessary OpenEdge and Java environment variables to where they are installed. After installing OpenEdge, you can use the command-line utility, Proenv, as needed, to access these environment variables.

Caution: Although editing environment variables is an option, this procedure is not recommended if Progress Version 8.3 and Version 9 (or WebSpeed Version 2.x and Version 3.x) products exist on the same system.

For more information on environment variables for OpenEdge, see the information on maintaining user environments in *OpenEdge Deployment: Managing ABL Applications* or your specific product documentation.

UNIX environment variables

This section describes the operating system-specific environment variables on a UNIX operating system.

For information about setting environment variables related to OpenEdge AppServer, OpenEdge WebSpeed, an OpenEdge DataServer, or the OpenEdge Adapter for SonicMQ, see *OpenEdge Application Server: Administration*, *OpenEdge Application Server: Developing WebSpeed Applications*, *OpenEdge Data Management: DataServer for Microsoft SQL Server*, *OpenEdge Data Management: DataServer for ORACLE*, or *OpenEdge Data Management: DataServer for ODBC*.

After installation, OpenEdge requires little if any additional configuration. However, some environment variables can be customized. As needed, you can access these environment variables using Proenv command-line utility.

Running the `Proenv` script sets `DLC` to this directory automatically. `Proenv` also adds `$DLC/bin` to your path and changes your current directory to the OpenEdge work directory set during installation.

You can edit the `.profile` of a user to set up environment variables automatically each time the user logs onto the system. Also, be sure to export environment variables to make them available to child processes.

This section describes the purpose of each environment variable. Usage with the Bourne shell is given, yet other shells use similar syntax.

[Table 8–1](#) describes the UNIX environment variables. Read these descriptions to determine the variables you want to set.

Notes: `$DLC` is an environment variable for the full pathname of the directory where OpenEdge is installed. You can run `Proenv` to automatically set `DLC` to this directory.

If you want to use a remote `DataServer`, you must set additional environment variables depending on the type of `DataServer` you want to use (for example, `ORACLE` or `ODBC`). See the `DataServer` documentation for more information on the other variables set.

When you first execute an OpenEdge command or utility that requires Java, OpenEdge correctly sets the Java environment variables based on your UNIX platform.

Table 8–1 identifies UNIX variables. In most instances, these variables are set by scripts and do not need user involvement.

Table 8–1: UNIX environment variables (1 of 5)

Variable	Description	Code example
DLC	The pathname of the directory where you installed the OpenEdge software. The default value is /usr/dlc. You must set this variable if you install the OpenEdge software in an alternate directory.	DLC=/usr/dlc
PATH	<p>A list of the directories UNIX searches to find any commands that you provide. OpenEdge also searches these directories for UNIX commands or programs you name when using the INPUT THROUGH and OUTPUT THROUGH statements:</p> <ul style="list-style-type: none">• Include \$DLC/bin in the PATH environment variable. To keep end users out of the /DLC directory, you can provide scripts to perform all OpenEdge-related actions. These scripts can reside somewhere else in the PATH and invoke OpenEdge commands with full pathnames. Place your startup, shutdown, and maintenance scripts somewhere in the path directories.• Include \$JAVAHOME/bin in the PATH environment variable. This value must be set to ensure that the installation can detect a java installation because the java executable is located in the \$JAVAHOME/bin. <p>Note: If during installation you chose yes to copy scripts to /usr/bin, ensure that PATH is set to /usr/bin: PATH=:/usr/bin.</p>	<p>PATH=\$PATH:\$DLC/bin</p> <p>PATH=\$JAVAHOME/bin</p>

Table 8–1: UNIX environment variables

(2 of 5)

Variable	Description	Code example
PROCFG	The filename (or full pathname) of your product's configuration file. The configuration file is a data file that identifies the OpenEdge product and components that you are licensed to use. The default value is \$DLC/progress.cfg. Reset PROCFG if you have moved your configuration file from the directory where you installed OpenEdge.	PROCFG=\$DLC/products.cfg
PROCONV	The filename (or full pathname) of the OpenEdge convmap.cp file. The convmap.cp file is a binary file that contains all of the conversion tables that are available to OpenEdge. The default value is \$DLC/convmap.cp. See <i>OpenEdge Development: Internationalizing Applications</i> for more information on the convmap.cp file.	PROCONV=\$DLC/convmap.cp
PROEXE	The pathname of your OpenEdge executable file. The default value is \$DLC/bin/_progres. If you move _progres out of \$DLC/bin, rename _progres, or use the OEBuild utility to generate a customized module, set PROEXE appropriately (or modify your scripts).	PROEXE=\$DLC/bin/_progres
PROLOAD	The pathname of the directory where you installed the OEBUILD product, if you installed it. The default value is \$DLC/oebuild. For example, if you installed OEBUILD in the directory /vol1/devdir/load, use the code example.	PROLOAD=/vol1/devdir/load
PROMSGS	The full pathname of your OpenEdge run-time messages file. The default value is \$DLC/promsgs. For example, if you want to use the German run-time messages file, use the code example in your profile. You only set the PROMSGS environment variable if you want to use a run-time messages file different from the default PROMSGS file in the \$DLC directory.	PROMSGS=\$DLC/prolang/ger/promsgs.ger

Table 8–1: UNIX environment variables

(3 of 5)

Variable	Description	Code example
JDKHOME	Establishes the top-level directory for the Java Developer's Kit (JDK). Note: When you first execute an OpenEdge command or utility that requires Java, OpenEdge correctly sets the Java environment variables based on your version of UNIX.	JDKHOME=\$DLC/jdk
JREHOME	Establishes the top-level directory for the Java Runtime Environment (JRE).	JREHOME=\$DLC/jre
JFCHOME	Establishes the top level directory for the Java Foundation Classes (JFC).	JFCHOME=\$DLC/jfc
JFCCP	Sets the classpath for Java Foundation Classes (JFC) only.	JFCCP=\$JFCHOME/swingall.jar
CLASSPATH	OpenEdge correctly sets the appropriate classpath variable based on the platform in use.	CLASSPATH=\$JDKCP:\$JFCCP:\$PROGRES SCP
JIT	Sets the just-in-time compiler correctly.	JIT="-nojit"
JVMEXE	Sets the Java Virtual Machine to run correctly.	JVMEXE=jre

Table 8–1: UNIX environment variables

(4 of 5)

Variable	Description	Code example
PROPATH	<p>A list of directories OpenEdge searches to find procedures.</p> <p>OpenEdge AppServer and OpenEdge WebSpeed use the PROPATH property in \$DLC/properties/ubroker.properties.</p> <p>Otherwise, by default, OpenEdge searches these subdirectories, using the order specified:</p> <ol style="list-style-type: none"> 1. \$DLC/tty 2. \$DLC 3. \$DLC/bin <p>Use the following syntax to set the PROPATH environment variable:</p> <pre>PROPATH=[:] {<i>dir-name</i> / <i>env-var</i>} (:...);</pre> <p>Where:</p> <p>[:]</p> <p>Tells OpenEdge to search your working directory before searching any other directories.</p> <p><i>dir-name</i></p> <p>Specifies the name of a directory that you want OpenEdge to search.</p> <p><i>env-var</i></p> <p>Specifies the environment variable whose definition names one or more directories that you want to search.</p> <p>(:...)</p> <p>Separates multiple <i>dir-name</i> or <i>env-var</i>.</p> <p>;</p> <p>Ends the definition of the PROPATH environment variable and indicates the start of a new command.</p>	PROPATH=:persapp:\$DLC

Table 8–1: UNIX environment variables*(5 of 5)*

Variable	Description	Code example
PROSRV	The pathname of your executable PROSERVE file. The default value is \$DLC/bin/_mprosrv. The PROSERVE script includes the code example. Therefore, if you move _mprosrv out of \$DLC/bin, rename _mprosrv, or use the OEBuild utility to create a customized module, set PROSRV appropriately (or modify your proserve script).	PROSRV=\$DLC/bin/_mprosrv
PROSTARTUP	The pathname of the OpenEdge default startup parameter file, startup.pf. This file is read by all OpenEdge modules at startup; it must exist for OpenEdge to execute properly.	PROSTARTUP=\$DLC/startup.pf
PROTERMCAP	The full pathname of the terminal definition file that you want your OpenEdge session to use. The default terminal definition file is called PROTERMCAP and is installed by default in the /\$DLC directory. You only have to set the PROTERMCAP environment variable if you want to use a different terminal definition file from the default PROTERMCAP file.	PROTERMCAP=\$DLC/SPECIALCAP
TERM	The type of terminal you are using. For example, to define your terminal type as wy75, use the code example.	TERM=wy75

Setting Java environment variables

By default, the OpenEdge installation program tailors all the necessary OpenEdge and Java environment variables to the directories where they are installed. For example, the installation automatically sets the %DLC% environment variable to your OpenEdge installation path.

OpenEdge bundles the Java Runtime Environment (JRE) component and the Java Development Kit (JDK) component with certain products that you install. For more information, see the “[Java considerations](#)” section on page 1–2. For additional details regarding Java and platform-specific information, see the “[Java requirements](#)” section on page 2–2. And, for more information on Java environment variables for OpenEdge products, see your specific product documentation.

Setting the JDK environment variable

In most circumstances, you will not need to set the JDK environment variable. When you load your installation medium, the Installation Program determines whether JVM is on your machine. It also verifies that you have the correct JVM version required to run OpenEdge. See “[Loading the installation media](#)” section on page 5–2 for more information. Only if needed, you can correctly set up your JDK environment for products that rely on the environment variables set by the script file \$DLC/bin/java_env (for example, JDKHOME and JREHOME).



To correctly set up your JDK environments if this task was not accomplished when your installation medium was loaded, you must edit this file and change the JDKHOME value from:

```
#JDKHOME=
```

To:

```
JDKHOME=/usr1/jdk-directory
```

Where /usr1/jdk-directory is the JDK install directory.

Note: This modification applies to the following sections of the \$DLC/bin/java_env file: HP-UX and HP Tru64. The JDK is bundled on the Sun Solaris platform, and therefore not needed. The root directory owns the java_env file, and the individual modifying the file must have root access.

Setting SQL client environment variables

SQL client environment variables are automatically set for you in `$DLC/bin/sql_env`.

Using the Proenv utility

The `$DLC` environment variable is not set at the system level. The Proenv utility can automatically set the `$DLC` environment variable to the directory where OpenEdge is installed. It then adds `$DLC/bin` to your PATH.

Proenv opens a new window, sets the environment variables, and then changes the current directory to the working directory you set when you installed OpenEdge.

Getting started with the AdminServer

The AdminServer is the key controlling element of the Progress Explorer Framework. It facilitates the tasks associated with managing and configuring your installation.

This section briefly introduces the AdminServer. For details about the AdminServer, its role in the Progress Explorer Framework, and the tasks to use the AdminServer, see [Chapter 10, “Configuration.”](#)

OpenEdge products supported by the AdminServer

An AdminServer is installed on every system where you install any of the following OpenEdge products:

- OpenEdge Databases: OpenEdge Personal RDBMS, OpenEdge Workgroup RDBMS, and OpenEdge Enterprise RDBMS.
- DataServers: OpenEdge DataServer for ORACLE and OpenEdge DataSever for MS SQL Server, and DataServer for ODBC.
- OpenEdge Application Server—Basic.
- OpenEdge Application Server—Enterprise.
- OpenEdge Studio.
- OpenEdge Architect.

- Web Services Adapter (AIA).
- AppServer Internet Adapter.
- OpenEdge Adapter for SonicMQ.
- OpenEdge Adapter for Sonic ESB.
- NameServer.
- OpenEdge NameServer Load Balancer.
- WebSpeed Workshop.
- OpenEdge Replication.
- OpenEdge Replication Plus.

A command-line utility, PROADSV, supports OpenEdge administrative capabilities on UNIX. PROADSV allows you to start up, shut down, and query the status of the AdminServer. See the sections about the PROADSV command in [Chapter 10, “Configuration”](#) for detailed syntax information.

AdminServer considerations

Note the following points that pertain to AdminServer usage:

- Before you start a WebSpeed or AppServer application, you must start the AdminServer.
- The AdminServer User-Group Authorization feature requires that you have privileges set to allow you access and operational privileges for the AdminServer. See the [“How to implement the User-Group Authorization feature”](#) section on page 8–12.
- The AdminServer must be running before you can use the OpenEdge Explorer, from a remote Windows machine, to configure and manage your applications.

How to implement the User-Group Authorization feature

To implement the User-Group Authorization feature on a UNIX platform, you must first successfully complete the OpenEdge installation program.

Table 8–2 identifies and briefly describes the purpose of each new command-line option.

Table 8–2: User-Group parameter options

Parameter name	Syntax	Purpose
Individual user name and password required.	-requireusername	Requires a minimum of one user ID to be resolved for each AdminServer operation before it can be executed.
Group authorization required.	-admingroup group [{, :}group...]	Requires a minimum of one group to be resolved for each AdminServer operation before it can be executed. A colon-separated list differentiates groups when you are specifying multiple groups on the command line.

On UNIX platforms, a group name can be any user-defined or NIS group name. UNIX can also support subgroups.

Understanding the built-in terminal definitions

This section provides a list of built-in terminal definitions supplied with OpenEdge as described in the following sections:

- [Terminal issues](#)
- [Terminal identifiers](#)

Terminal issues

When you start OpenEdge, you might receive the following message:

```
** You cannot use DEL for both stty intr and DELETE-CHARACTER.
```

The message as presented in the previous message indicates that you were trying to use the **DEL** key as the UNIX interrupt key and as the OpenEdge **DELETE-CHARACTER** key. To avoid this message, add the following line to your `.profile` file:

```
stty intr ^C
```

This command resets your UNIX interrupt key from **DEL** to **CTRL+C**.

Built-in terminal definitions are supplied with OpenEdge for the terminals listed in [Table 8-3](#), which indicates the terminal identifiers you can use so that OpenEdge can successfully access that terminal definition. Be sure the operating system environment variable `TERM` is set to the appropriate value. For example:

```
TERM=wyse60;export TERM
```

Terminal identifiers

Table 8–3 shows a list of terminal identifiers.

Table 8–3: Terminal identifiers (1 of 2)

Terminal model	Terminal identifier	Notes
xterm	xterm	–
CDE dtterm	dtterm	–
DEC VT100	V1, vt100, vt100-80	Asian languages are supported. For more information on supported languages, see <i>OpenEdge Development: Internationalizing Applications</i> .
DEC VT200 series	V2, vt200, vt200-80, vt220, vt220-80, vt240, vt241	Asian languages are supported. For more information on supported languages, see <i>OpenEdge Development: Internationalizing Applications</i> .
DEC VT300 series	V3, vt300, vt320, vt330, vt340, pt300, pt-100	–
DEC VT400 series	V4, vt400, vt400-80, vt420	–
DEC VT500 series	V5, vt500, vt500-80, vt510, vt520, vt525	–
IBM 3151	3151, m3, ibm3151	–
IBM PC/AT XENIX console	li, ansi	Ansi driver.
IBM PC/AT XENIX color console	lc, ansic	Ansi driver. Uses reverse video for input fields.
Linux console	linux, linux-lat	–
Sun console	Mu, sun	–

Table 8–3: Terminal identifiers (2 of 2)

Terminal model	Terminal identifier	Notes
Wyse 60	w6, wy60, wyse60 wy60t, wyse60tall wy60w, wyse60wide wy60tw, wyse60tall + wide	Assumes that the function keys are set to the factory defaults. Check the PROTERMCAP entry for setup mode. Terminal in following mode: 43 lines X 80 columns. 25 lines X 132 columns. 43 lines X 132 columns.
Wyse 370	wy370, wyse370	—

Additional terminal identifier considerations

Note the following points related to terminal identifiers:

- [Table 8–3](#) is complete as of the print date of this guide.
- The IBM native console terminal type `hft` is not supported.
- OpenEdge does not support spacetaking terminals, unless the terminal has a firmware setup option to change it to a nonspacetaking mode.



To determine your terminal identifier if it is not listed in [Table 8–3](#):

1. Try to run OpenEdge using a terminal definition for a terminal that functions similarly to yours, or try to configure your terminal to emulate one of the supported terminals.

Note: Progress Software Corporation does not support terminal emulation.

2. In the directory where you installed your OpenEdge product, find the PROTERMCAP file that contains terminal definitions.

Note: All terminal types supported by OpenEdge are documented in the \$DLC/protermcap file.

3. Search through the PROTERMCAP file to see if your terminal is listed. The PROTERMCAP file is similar in structure to the UNIX /etc/termcap file. Each terminal type is followed by a description of that terminal. For more information about the PROTERMCAP file, see [OpenEdge Deployment: Managing ABL Applications](#).

OpenEdge key and certificate stores

All OpenEdge server and client components that implement Secure HTTP (HTTPS) or Secure Socket Layer (SSL) connections require access to private keys and digital certificates to negotiate these connections and to enable them to function securely.

For all OpenEdge components, OpenEdge provides utilities that allow you to install and manage keys and digital certificates (in key stores and certificate stores) so the components can use them. For Open Clients and clients of Web services, OpenEdge provides utilities for some clients or relies on utilities provided by the client platform to manage the required certificate stores.

For more information on managing certificate stores for Open Clients and Web service clients, see [OpenEdge Development: Open Client Introduction and Programming](#). For details about using the OpenEdge utilities to manage key stores for OpenEdge servers and manage certificate stores for OpenEdge clients, see [Chapter 9, “Managing OpenEdge Key and Certificate Stores”](#) and [Appendix C, “Command and Utility Reference.”](#)

Managing OpenEdge Key and Certificate Stores

All OpenEdge server and client components that implement Secure HTTP (HTTPS) or Secure Socket Layer (SSL) connections require access to private keys and digital certificates to negotiate these connections and to enable them to function securely.

For all OpenEdge components, OpenEdge provides utilities that allow you to install and manage keys and digital certificates (in key stores and certificate stores) so the components can use them. For Open Clients and clients of OpenEdge Web services, OpenEdge provides utilities for some clients or relies on utilities provided by the client platform to manage the required certificate stores.

This chapter describes how to use the OpenEdge utilities, as detailed in the following sections:

- [Managing key stores for OpenEdge servers](#)
- [Managing certificate stores for OpenEdge clients](#)

An SSL server requires access to a private key and digital (public-key) certificate to authorize the identity of the server. Clients require access to public-key certificates that allow them to authenticate the servers that they access. Both servers and clients must obtain their keys and certificates from a trusted source, a Certificate Authority (CA). The server can trust the CA to authorize the server's identity and the client can trust the CA to provide proof of the server's identity. For more information on keys, certificates, and how CA's support them, see the chapters on security in *OpenEdge Getting Started: Core Business Services*.

Managing key stores for OpenEdge servers

You can manage the private keys and corresponding digital certificates for OpenEdge servers that support SSL connections using a *key store* located in the *OpenEdge-Install-Dir\keys* directory. Each SSL server requires at least one *key store entry* that contains a single private key and corresponding digital (public-key) certificate. With this key store entry, you can configure any supported OpenEdge server to enable and manage SSL connection from clients. For more information on the OpenEdge servers that support SSL server configuration, see the sections on the OpenEdge-supported SSL server components described in *OpenEdge Getting Started: Core Business Services*.

If you require only data encryption and do not need to verify the identity of SSL servers (typically, for intranet configurations only), OpenEdge comes installed with a default key store entry. This default entry contains a common private key and digital certificate pair that you can use without any further management beyond enabling SSL connections on OpenEdge clients and servers. For more information on the default SSL server identity, see the sections on SSL in OpenEdge in *OpenEdge Getting Started: Core Business Services*.

However, to establish a trusted OpenEdge SSL server identity suitable for use on the Internet or a more secure intranet, you must complete several steps using the functions of the `pkiutil` command-line utility installed with OpenEdge.

Notes: Before you run an OpenEdge command-line utility, set the DLC environment variable to the *OpenEdge-Install-dir* pathname and set the `WRKDIR` environment variable to your working directory. For an example, see the *OpenEdge-install-dir/bin/pkiutil* shell script on UNIX or the *OpenEdge-install-dir\bin\pkiutil.bat* file in Windows.

Running the command-line utility in a `Proenv` command window properly sets `DLC` and `WRKDIR` for you.

Establishing a trusted SSL server identity

There are several steps required to establish a trusted identity for any OpenEdge SSL server using the `pkiutil` command-line utility.

Caution: While the `default_server` key store entry provided by the Progress Server Certificate Authority also uses a default password ("password"), you must password-protect any private key store entries that you create from a public-key certificate issued by a trusted external CA. The secrecy of your password is critical to using this key store entry for authenticating a server.

Steps for creating and managing a key store entry



To establish and maintain a trusted SSL server identity using the `pkiutil` utility:

1. Use the `-newreq` operation to generate a proposed public and private-key pair together with a digital certificate request that is suitable for sending to any CA for authorization. You must provide a password to secure this certificate request. You must later provide this password to any OpenEdge server that you want to access this key store entry for securing SSL connections to it. See the [“Supplying a key store entry password to an OpenEdge server”](#) section on page 9–4.
2. Use e-mail, or some other method required by the CA, to send a copy of the certificate request to the trusted CA you want to return a public-key certificate that can authenticate any server that you provide access to the private key.
3. Use the `-import` operation to import the digital certificate returned by the CA for this request and store it together with the associated private key as an entry in the key store.
4. Use the `-display` or `-list` operations to review an individual digital certificate file or any and all key store entries for important digital certificate information, such as expiration dates.
5. Use the `-remove` operation to remove any unused or expired key store entries that you specify and retain them in a backup area of the key store.

For an overview of the `pkiutil` command-line utility, see the [“Using pkiutil to manage an OpenEdge key store”](#) section on page 9–5.

Supplying a key store entry password to an OpenEdge server

When you configure an OpenEdge server to access a key store entry, you must provide it with the same password that you used to create the key store entry. If you configure the server using the Progress Explorer, you can enter this password directly in the fields provided. However, if you configure the server by manually editing the `ubroker.properties` file for that server or specifying the password on a command line or in a startup script (as required when starting a database server for the OpenEdge RDBMS), you must provide an encrypted value for the password in order to protect the password itself from being easily discovered. OpenEdge provides the `genpassword` command-line utility for obtaining a password's encrypted value. For more information, see the [“Using genpassword to obtain a key store password-encrypted value”](#) section on page 9–8.

Using pkiutil to manage an OpenEdge key store

The `pkiutil` command-line utility provides all the operations necessary to create and manage key store entries for OpenEdge SSL servers (See the [“Managing key stores for OpenEdge servers”](#) section on page 9–2). This utility manages all input and output for the key store in the `OpenEdge-Install-Dir\keys` directory. For more information on the structure of this directory, see the [“Understanding key store content”](#) section on page 9–6.

The `pkiutil` utility has the following general command-line syntax:

```
pkiutil [ options ] function arguments
```

options

Change the type of information and defaults for different functions (*function*) of the utility.

function arguments

One of the following functions (*function*) and the objects they affect (*arguments*):

- **`-newreq alias`** — Generates a new private/public-key pair and a corresponding public-key certificate request (suitable for submission to a CA), stored under the alias name specified by *alias*.
- **`-import alias cert-file`** — Imports a CA-issued SSL server digital (public-key) certificate from the disk file, *cert-file*, pairs it with the private key generated for a public key request identified by the alias name, *alias*, and places the pair in the key store as a new entry identified by *alias*.
- **`-print alias`** — Displays the public-key certificate request identified by *alias*.
- **`-list [alias . . .]`** — Displays a list of specified (*alias*) or all current key store entries.
- **`-display cert-file`** — Displays the digital certificate file information contained in the operating system disk file, *cert-file*.
- **`-remove alias . . .`** — Removes one or more specified (*alias*) key store entries.

For complete information on the options and functions of the `pkiutil` command-line utility, see [Appendix C, “Command and Utility Reference.”](#)

Understanding key store content

The OpenEdge key store maintains private keys and digital certificates for OpenEdge SSL servers in several locations. These include private keys and digital certificates that you have authorized by a CA and imported for use by an SSL server, and private keys and public-key certificate requests that you generate and have pending for authorization by a CA. You must manage this key store entirely with the `pkiutil` command-line utility. See the [“Using pkiutil to manage an OpenEdge key store”](#) section on page 9–5 for additional information.

The key store resides in the *OpenEdge-Install-Dir\keys* directory. This directory contains the following files and subdirectories:

- **alias.pem** — Files containing a single key store entry that you have created from an imported CA-authorized digital certificate that contains the public key joined with the private key that you generated along with the original public-key certificate request. Each file is named with the *alias* that you chose for the original private key and certificate request using the `-newreq` operation of `pkiutil`. The initial key store entry is the default OpenEdge entry, `default_server.pem`, as authorized by the Progress Software Corporation CA. For more information on this default key store entry, see the sections on SSL in *OpenEdge Getting Started: Core Business Services*.
- **policy** — A subdirectory containing a `pscpk.conf` configuration file. The `pkiutil` utility uses this file to control the process of generating new SSL server private/public keys and generating digital certificate requests that can be sent to a CA in order to obtain a public-key certificate for the OpenEdge SSL server. Initially, this is the only subdirectory.
- **requests** — A subdirectory containing all newly generated private keys and public-key certificate requests in the form of two files, as follows:
 - **alias.pk1** — This file holds the PKCS #1-formatted, password-encrypted, private key for the given key store *alias* entry.
 - **alias.pk10** — This file holds the PKCS #10-formatted public-key certificate request that you send to a CA to obtain the SSL server’s public-key certificate for the given key store *alias* entry.

- **backup** — A subdirectory containing any removed key store entries. The `pkiutil` utility removes an existing key store entry when you:
 - Explicitly remove it using the `-remove` operation of `pkiutil`.
 - Update an existing key store entry with a new digital certificate. You will perform this operation when the previous public-key certificate has expired and you have applied to the CA for a renewed public-key certificate.

In all cases, `pkiutil` places removed key store entries in this directory in case you find it necessary to recover and use them again.

Note: Performing successive `-remove` or `-import` operations on the same key store entry repeatedly overwrites that entry in the backup subdirectory.

Caution: If you upgrade or uninstall OpenEdge, Progress Software Corporation recommends that you back up your current version of the OpenEdge key store directory tree (*OpenEdge-Install-Dir\keys*) to prevent losing valuable keys and certificates.

Using genpassword to obtain a key store password-encrypted value

When you must configure an OpenEdge SSL server by manually editing the `ubroker.properties` file, or for the OpenEdge RDMS when you start up the database server to enable SSL connections, you must specify the password to allow access to the required private-key alias. The value you specify is available to anyone who can read the file or command line where you enter it. In order to prevent access to this password by unauthorized users, you must specify an encrypted form of the password that is equivalent to the password itself.

Note: You must also provide the encrypted form of the password ("password") for the `default_server` alias. In the Progress Explorer, when you configure an SSL server with the `default_server` alias, OpenEdge automatically provides the encrypted form of this password.

OpenEdge provides the `genpassword` command-line utility that you can use to obtain the encoded and encrypted form for the real password.

For example, when the following code is executed in the OpenEdge Proenv command window, you can generate an encrypted value for a password whose value is "topsecret":

```
proenv>genpassword -password topsecret
243d3f343726213624

proenv>
```

Later, to verify that an existing encrypted value matches the real password value, you can run `genpassword`, as follows:

```
proenv>genpassword -password topsecret -verify 243d3f343726213624
The passwords match.

proenv>
```

For more information on the options of the `genpassword` command-line utility, see [Appendix C, "Command and Utility Reference."](#)

Managing certificate stores for OpenEdge clients

You can manage trusted CA/root digital (public-key) certificates for OpenEdge clients that support SSL connections using a *root certificate store* located in the *OpenEdge-Install-Dir\certs* directory. Each OpenEdge SSL client requires the *root certificate store entry* that contains the public-key certificate from the CA who signed and issued the public-key certificate for the SSL server that the client needs to access. Without access to this CA's root digital certificate the OpenEdge client will be unable to validate the identity of the SSL server and will abort the SSL connection process. For more information on the OpenEdge client components that support SSL client configuration, see the documentation on the supported SSL client components described in *OpenEdge Getting Started: Core Business Services*.

If you require only data encryption and do not need to verify the identity of SSL servers (typically, for intranet configurations only), OpenEdge comes installed with the root digital certificate from the Progress Software Corporation CA (who also signed and issued the default_server key store digital certificate for OpenEdge SSL servers). The Progress Software Corporation CA root digital certificate is distributed in PEM format as d9855a82.0 and in DER format as pscca.cer (suitable for importing into a Windows workstation for use by an OpenEdge .NET Open Client). This default entry contains a common root public-key certificate that you can use to access any supported OpenEdge SSL server. For more information on the default root public-key certificate, see the sections on the OpenEdge default server identity in *OpenEdge Getting Started: Core Business Services*.

Installing trusted CA/root certificates

To allow OpenEdge client access to an SSL server whose identity you need to have verified, you must install the appropriate root digital certificate to authenticate that server. An SSL server can have its identity established from one of two basic sources:

- One of the trusted public CA root digital certificates distributed by Progress Software Corporation that includes RSA, Thawte, and Verisign.
- A root digital certificate from an internal CA that you have set up on your own certificate server or from another external or public CA other than RSA, Thawte, or Verisign.

OpenEdge automatically installs root certificates in the OpenEdge root certificate store from RSA, Thawte, and Verisign. However, if you use your own internal-use CA or a public CA other than these three, you must install the required root certificates yourself.

OpenEdge provides the following command-line utilities to install and manage root certificates in the OpenEdge certificate store:

- **certutil** — Installs, lists, and manages CA/root certificates from any CA as entries in the OpenEdge root certificate store and manages the certificate store for the client. You can also remove certificate store entries using this utility. The utility moves all removed entries to a backup subdirectory of the root certificate store for future recovery and use.

Note: For .NET and Java Open Clients and Web service clients of OpenEdge application servers, you must use other utilities to manage the root certificate stores for those clients. For more information, see *OpenEdge Development: Open Client Introduction and Programming*.

- **mkhashfile** — Provides simple installation of PEM-encoded root certificates into the OpenEdge root certificate store from any CA, but provides no other management functions for the OpenEdge certificate store. You can use `certutil` for the additional root certificate management.

Notes: Before you run an OpenEdge command-line utility, set the DLC environment variable to the *OpenEdge-install-dir* pathname and set the `WRKDIR` environment variable to your working directory. For an example, see the *OpenEdge-install-dir/bin/pkiutil* shell script on UNIX or the *OpenEdge-install-dir\bin\pkiutil.bat* file in Windows.

Running the command-line utility in a Proenv command window properly sets DLC and `WRKDIR` for you.

Using certutil to manage an OpenEdge root certificate store

The `certutil` command-line utility provides functions to install root certificates from any CA and to manage all of the entries in the OpenEdge root certificate store.

The `certutil` utility has the following general command-line syntax:

```
certutil [ options ] function arguments
```

options

Change the type of information and defaults for different functions (*function*) of the utility.

function arguments

One of the following functions (*function*) and the objects they affect (*arguments*):

- **-import cert-file** — Imports a trusted CA root certificate from the disk file, *cert-file*, and creates a root certificate store entry identified by a generated alias name (*alias*, as specified for other functions of this utility).
- **-list [alias . . .]** — Displays a list of specified (*alias*) or all current certificate store entries.
- **-display cert-file** — Displays the digital certificate file information contained in the operating system disk file, *cert-file*.
- **-remove alias . . .** — Removes one or more specified (*alias*) certificate store entries.

For more information on the options and functions of the `certutil` command-line utility, see [Appendix C, “Command and Utility Reference.”](#)

Using mkhashfile to install root certificates in the OpenEdge root certificate store

The `mkhashfile` command-line utility provides a simple way to install a root certificate that is authorized by your own internal-use CA, or any CA that can provide you with a PEM-encoded certificate (typically in a file named with the `.pem` extension). If you are using your own certificate server to provide the certificate, see the documentation for the certificate server administration software for information on how to obtain PEM-encoded certificates. Once you have the certificate accessible to your OpenEdge SSL client machine, you can use the `mkhashfile` command-line utility to install it in the OpenEdge root certificate store.

Note: If the root certificate is not a PEM-encoded certificate, it is recommended that you use the `certutil` command-line utility, specifying the format option. For details about the `certutil` command-line utility and all its options and functions, see the detailed syntax information for the `certutil` command listed in [Appendix C, “Command and Utility Reference.”](#)

To use `mkhashfile` to create an entry in the OpenEdge root certificate store for a local PEM-encoded certificate file, `vsigntca.pem`, specify the file with the `mkhashfile` command that you enter in the OpenEdge Proenv command window. For example:

```
proenv>mkhashfile vsigntca.pem

OpenEdge Release 10.0B02 as of Sat Feb  5 00:15:12 EST 2005

Running SSLC command ...
Copying vsigntca.pem and 18d46017.0 to C:\Progress\OpenEdge\certs
proenv>
```

The utility generates the entry as a file with an encrypted filename, `18d46017.0`, which is the alias used to identify the certificate store entry. You can then manage this entry along with all other entries in the OpenEdge certificate store using the `certutil` utility. For more information, see the “[Using certutil to manage an OpenEdge root certificate store](#)” section on page 9–11.

For more information on the `mkhashfile` command-line utility, see [Appendix C, “Command and Utility Reference.”](#)

Configuration

Once you have installed OpenEdge, you can perform configuration tasks as needed to support your application goals. This chapter introduces the Progress Explorer Framework, a common administrative architecture for installed OpenEdge server products, and highlights the framework's elements, focusing on Unified Brokers, as presented in the following sections:

- [Overview of the Progress Explorer Framework](#)
- [Working with Unified Brokers](#)
- [Understanding and Using the AdminServer](#)
- [Using the Progress Explorer tool](#)
- [Mergeprop utility overview](#)
- [Ubroker.properties file and product configurations](#)
- [Command-line utilities reference](#)

Overview of the Progress Explorer Framework

This section:

- Provides a brief introduction to the Progress Explorer Framework.
- Identifies some of the OpenEdge products that use the framework's functionality, focusing on Unified Brokers.
- Presents a conceptual diagram of the Progress Explorer Framework and detailed descriptions of its elements.

Introduction

The Progress Explorer Application Framework is a system administration framework that provides a consistent interface in which specific OpenEdge products can be managed. The framework's elements support common administrative tasks and activities you can use to start and stop processes, and to manage, configure, and validate properties for specific OpenEdge products. The AdminServer process, the key element of the framework, enables supported products to address their specific requirements. The AdminServer also supports various management utilities to provide similar configuration and management functions for all of these products. For a complete list of products that use the AdminServer, see the [“OpenEdge products supported by the AdminServer”](#) section on page 7–12.

Some OpenEdge products administered through and managed by the Progress Explorer Framework are designed to help manage an application's resources. For example, these products are based on receiving and sending requests through brokers. Brokers poll for available resources (that is, client and agents), attempting to fulfill these requests. The Progress Explorer Framework facilitates the common administrative tasks and configuration activities that are fundamental to the technology these broker-based products use.

OpenEdge products that support broker functionality include:

- **Unified Brokers** — WebSpeed, AppServer, DataServer for MS SQL Server, Oracle DataServer, and ODBC DataServer.
- **Adapters** — AppServer Internet Adapter, Web Services Adapter, and OpenEdge Adapter for SonicMQ.
- **Messengers** — CGIIP, WSASP, WSISA, and WNSNA.

Data for each Unified Broker product is stored in a common text file called `ubroker.properties` file. The file stores the property and configuration information for each Unified Broker. The Progress Explorer Framework supports tools like the `mergeprop` utility and the Progress Explorer tool that you use to manage the contents of these files.

This chapter focuses on the Unified Brokers and how the Progress Explorer Framework supports them to manage an application's resources and make these resources available to clients.

Note: The OpenEdge database is another key product that is part of the Progress Explorer Framework. In contrast to the Unified Brokers and their relationship to the `ubroker.properties` file, all configuration changes made to any database administered through the AdminServer are stored in the Configuration Manager properties (`conmgr.properties`) file. For information on OpenEdge database administration, see [OpenEdge Data Management: Database Administration](#).

Framework elements and descriptions

[Figure 10–1](#) shows the conceptual relationship among the framework's elements, products, and tools. [Table 10–1](#) identifies each element of the Progress Explorer Framework, briefly defines the element, and points to other sections within this document or the OpenEdge documentation set where additional information about an element is presented.

Although all Unified Brokers are controlled through the same Unified Broker Properties file (`ubroker.properties`), each broker type maintains a unique port separate from any other server in the group. Any configuration change made to a Unified Broker administered through the AdminServer is stored in the `ubroker.properties` file.

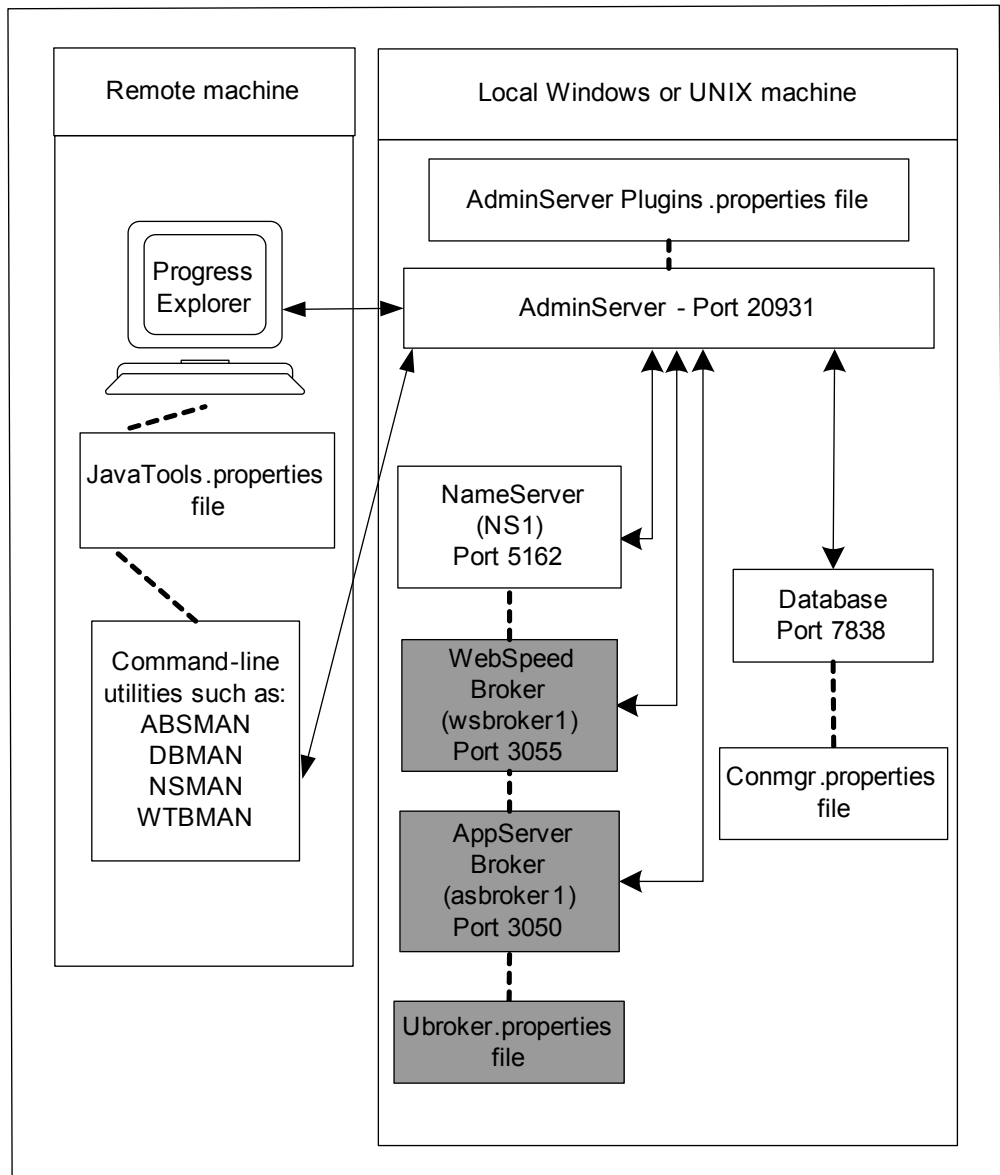


Figure 10-1: Overview of the Progress Explorer Framework

The shaded elements in Figure 10-1, WebSpeed and AppServer, are two of the Unified Brokers products. They are intended to represent all Unified Broker products in this graphic to show how the Unified Brokers relate to the other framework's elements.

Table 10–1 highlights and describes each element of the Progress Explorer Framework.

Table 10–1: Elements of the Progress Explorer Framework

(1 of 4)

Element	Description	For more information about this element, see . . .
AdminServer ¹	<p>As the central control within the framework, the AdminServer:</p> <ul style="list-style-type: none"> • Manages each instance of an installed OpenEdge server² product by providing administrative access to OpenEdge products installed on your network. • Governs remote management and configuration capabilities. • Supports an administrative feature which users can set to limit access to OpenEdge products. 	<p>The “Getting started with the AdminServer” section on page 7–11 (Windows platforms), the “Getting started with the AdminServer” section on page 8–10 (UNIX platforms), and Appendix G, “AdminServer Authorization and Authentication.”</p>
Progress Explorer configuration tool ³	<p>A graphical user interface tool to:</p> <ul style="list-style-type: none"> • Initiate administrative tasks on local or remote machines that require a running AdminServer. • Perform a variety of administrative, managerial, configuration, and validation activities for OpenEdge products. • The Progress Explorer only runs in Windows. However, you can connect to remote UNIX systems and administer various supported components on those remote UNIX systems. 	<p>The “Using the Progress Explorer tool” section on page 10–23. Also, see the Progress Explorer online help for extensive information about using the tool.</p>

Table 10–1: Elements of the Progress Explorer Framework

(2 of 4)

Element	Description	For more information about this element, see . . .
Mergeprop utility ³	A command-line utility that supports functionality similar to that supported by the Progress Explorer configuration tool. The mergeprop utility can be used as an alternative approach when the Progress Explorer configuration tool is not available.	The “ Mergeprop utility overview ” section on page 10–26.
Command-line utilities	A basic command-line tool that allows you to control (that is, start, stop, and query) servers and validate property files associated with OpenEdge products.	The “ Command-line utilities reference ” section on page 10–43.
Unified Broker	<p>As the central control within particular OpenEdge products, the Unified Broker is:</p> <ul style="list-style-type: none">• The key process through which each of these products’ resources are individually managed by the product, and these resources are made available to clients.• A collective term used to identify specific OpenEdge products that employ the same mechanism to implement the broker processes.• A standard processing technology within certain OpenEdge products.	The “ Working with Unified Brokers ” section on page 10–9.

Table 10–1: Elements of the Progress Explorer Framework*(3 of 4)*

Element	Description	For more information about this element, see . . .
ubroker.properties file ³ (Unified Broker properties file)	<p>Common text file location in which data for each Unified Broker⁴ product is stored.</p> <p>The Progress Explorer’s and the mergeprop utility’s capabilities can be applied to the contents of the <code>ubroker.properties</code> file to manage, configure, or validate properties for each of these products.</p> <p>The <code>ubroker.properties</code> file is located in <code>OpenEdge-install-dir/properties /</code> directory.</p>	The “ Ubroker.properties file and product configurations ” section on page 10–38.
conmgr.properties file (Configuration Manager properties file)	<p>Common text file that contains configuration information for all OpenEdge databases⁵.</p> <p>The Progress Explorer’s and mergeprop utility’s capabilities can be applied to the contents of the <code>conmgr.properties</code> file to manage, configure, or validate properties for each of these products.</p> <p>The <code>conmgr.properties</code> file is located in <code>OpenEdge-install-dir/properties /</code> directory.</p>	The “ Ubroker.properties file and product configurations ” section on page 10–38. Also, see <i>OpenEdge Data Management: Database Administration</i> .

Table 10–1: Elements of the Progress Explorer Framework

(4 of 4)

Element	Description	For more information about this element, see . . .
AdminServer plugins.properties file	Common text file that contains information for plugins to be loaded and managed by the AdminServer. The AdminServer plugins.properties file is located in <i>OpenEdge-install-dir/properties</i> / directory.	The “ Changing the default port ” section on page 10–19.
JavaTools.properties file (OpenEdge clients’ configuration file)	Common text file that contains configuration information for all OpenEdge clients. The JavaTools.properties file is located in <i>OpenEdge-install-dir/properties</i> / directory.	Do not make user-modifications to these property files as these properties support OpenEdge and Progress products only. Contact Progress Technical Support if you want to modify these properties.

1. The AdminServer must be running to use the management command-line management utilities (such as ASBMAN, DBMAN, NSMAN, and WTBMAN) or the Progress Explorer configuration tools. Only mergeprop and the Progress Explorer configuration tool perform the actual configuration of Progress server products and such changes affect the data stored in the `ubroker.properties` or `conmrg.properties` files.
2. See the “[OpenEdge products supported by the AdminServer](#)” section on page 7–12 (Windows platforms) and the “[OpenEdge products supported by the AdminServer](#)” section on page 8–10 (Unix platforms) for specific products.
3. Commands entered and accepted either through the Progress Explorer tool or the mergeprop tool immediately affect the data stored in the `ubroker.properties` file or `conmrg.properties` file.
4. The Unified Broker products include these OpenEdge servers: AppServer, WebSpeed, NameServer, and the Oracle DataServer, the DataServer for MS SQL, and the DataServer for ODBC. See the specific book within the product documentation set for more information about each Unified Broker product.
5. Only those OpenEdge databases that are configured to autostart will start when the AdminServer starts.

Additional Progress Explorer Framework Considerations

In addition to the footnotes associated with [Table 10–1](#), note these points about [Figure 10–1](#):

- The AdminServer, through its default port 20931 as shown in the diagram, processes start, stop, and query requests initiated from a requesting OpenEdge product. Similarly, the database, NameServer, AppServer, and WebSpeed brokers have their own default ports.

As part of setting up and maintaining security measures for your machines, it is advisable to change default port numbers and server names. Doing so helps to protect the identifies of these ports from personnel outside your organization. Also, consider documenting and monitoring all of your ports (that is, the port numbers and types) that you use for the AdminServer.

- The dotted lines connecting the `ubroker.properties` file and `conmgr.properties` file to their respective OpenEdge products is intended to indicate that the commands entered and accepted either through the Progress Explorer tool or the mergeprop utility directly affect the data stored in the `ubroker.properties` file or `conmgr.properties` file.
- Ensure that you backup the `ubroker.properties` and `conmgr.properties` files periodically because they contain the detailed configuration data for each OpenEdge product.

Working with Unified Brokers

The Unified Broker products include a Unified Broker process that is the initial point of client connection to a Unified Broker product instance. This broker process is responsible for managing other process resources that are part of the product, and making those resources available to clients. For more information about the “[Unified Broker products and associated clients](#)” section on page 10–39.

A Unified Broker and its related components can be set up to run locally or remotely.

Running locally

When a Unified Broker is run locally, the Unified Broker and all of its components are on the same machine. All Unified Broker products require that these components reside on the same machine: the Unified Broker instance and associated processes, the AdminServer, and the `ubroker.properties` file.

Running remotely

When a Unified Broker is run remotely, some components are distributed on separate machines, but connected on the same network.

When a Unified Broker product is distributed remotely, a separate AdminServer and `ubroker.properties` file must exist on each machine for access by the Progress Explorer tool. And, for WebSpeed, the Unified Broker Client (that is, WebSpeed Messenger) also resides on the same machine as the Web server, and Web clients (that is, browsers) can reside anywhere on the Internet, intranet, or extranet serviced by the Web server.

Regarding a DataServer, the separate database host for a DataServer applies only to WebSpeed or the AppServer. For a DataServer, the Unified Broker host is the DataServer host. The location of the target database management system (DBMS)—residing on a separate database host or must reside on the same machine as the DataServer host—depends on the DataServer and its platform.

Therefore, you can distribute a Unified Broker instance and its management components among three separate machines on the same network.

Unified Broker common elements

Complete administration for a Unified Broker application potentially involve these components shown in [Figure 10-1](#):

- The shaded elements which represent individual Unified Broker products.
- The Progress Explorer tool.
- Command-line utilities.

The AdminServer unifies the components previous listed. For some components and administration tasks you can use AdminServer-based management utilities (including the Progress Explorer) or a text editor and configuration validation utilities to accomplish the task.

Note: Similarly, the NameServer resides on its own machine, is installed with an administration framework, including an AdminServer and its own `ubroker.properties` file. If you use a text editor to modify the `ubroker.properties` file, the editor and configuration utilities must reside on the same machine as the Unified Broker or NameServer instance, or have network file system access to the respective Unified Broker and NameServer installation files.

Using default sample brokers

Most Unified Broker products have a default, sample broker that is immediately available for you to use. The purpose of these brokers is to help you quickly become familiar with and use the functionality associated with these products.

In most instances, the sample brokers require little, if any, modification and no validation. Although you can continue to use these sample brokers when you are operational in either a production or development mode, it is not advisable to do so. Consider modifying these files, using the edit capabilities of such tools as the Progress Explorer or mergeprop utility, for the purposes of security and tailoring them to your exact needs. A sample broker by default is not connected to a database. If you elect to use a default, sample broker, you will need to modify it if you need a database connection.

Table 10–2 identifies each default, sample brokers associated with each Unified Broker product.

Table 10–2: Default sample broker for each Unified Broker product

This Unified Broker product . . .	Has as its broker type . . .	And also supports the default broker identified as . . .
AppServer	AppServer Broker	asbroker1
OpenEdge Adapter for SonicMQ	Adapter for SonicMQ Broker	sonicMQ1
WebSpeed	WebSpeed Transaction Broker	wsbroker1
DataServer for MS SQL Server	DataServer MS SQL Broker	mssbroker1
ORACLE DataServer	ORACLE DataServer Broker	odbbroker1
DataServer for ODBC	DataServer for ODBC Broker	orabroker1
Name Server	NameServer Broker	NS1

Each Unified Broker process, default or user defined, manages only one Unified Broker process instance of the same type.

For general information about configuring a Unified Broker process, see “[Configuring and starting Unified Broker instances](#)” section on page 10–13. For specific details on how to configure the Unified Broker process for a product, how clients specify connections, and how the Unified Broker manages connections with clients, see your product documentation.

Additional Unified Broker characteristics

The Unified Broker process:

- Can register the following information with a controlling NameServer:
 - The broker’s location on the network.
 - The weight factor that you specify for load balancing.
 - The Application Services that you specify.

Note: Keep in mind that the NameServer is not required. Therefore, the registration of a Unified Broker with the NameServer is dependent on your specific implementation. [Appendix E, “NameServer and Name Server Load Balancing Details,”](#) for more information about the NameServer and the Unified Broker and NameServer relationship.

- Manages connections between clients and the Unified Broker instance.
- Provides other services unique to a Unified Broker product. For example, it maintains the status of each ABL process running on an AppServer and scales the number of processes according to changing demand.

Configuring and starting Unified Broker instances

This section describes:

- The two preliminary tasks you must complete before you can begin configuring and operating a Unified Broker product.
- The general steps to configure and start up Unified Broker instances.

Prerequisites to configure and use Unified Broker products

There are two preliminary tasks you must complete before you can begin configuring and operating a Unified Broker product:

- **Configure all machines involved in product installation and operation** — This task depends on how you plan to distribute your product and its applications on a network. For more information on configuring OpenEdge products on a network, see [Appendix F, “Configuration Models.”](#)
- **Install the necessary product components** — Typically, this involves installing, on one or more network machines, the OpenEdge Unified Broker product and additional software components are required to use the product, such as OpenEdge client or Web server software. If you plan to configure fault-tolerant servers or use load balancing, you must install a product that includes load balancing or install the load-balancing option for your product.

For more information on the OpenEdge product installation, see [Chapter 3, “OpenEdge Installation Prerequisites,”](#) either [Chapter 4, “Performing an OpenEdge Installation in Windows,”](#) or [Chapter 5, “Performing an OpenEdge Installation on UNIX,”](#) and the Windows- or UNIX-specific online help. For more information on the distributing resources in a Unified Broker environment, see the [“Working with Unified Brokers”](#) section on page 10–9.

Once you complete these preliminary tasks, you can configure and start up Unified Broker instances.

How to configure and start up Unified Broker instances

The following procedure presents the general steps required to configure and start up Unified Broker instances. Although much of this information has previously been presented in this chapter's earlier sections, it is helpful to have a general outline of the configuration and startup activities.

The properties file that comes installed with your Unified Broker product includes one sample Unified Broker and NameServer instance for each type of Unified Broker that you can use as a guide.



To configure and start up Unified Broker instances:

1. Start the AdminServer process on the machine on which each Unified Broker is installed:

- In Windows, OpenEdge installs the AdminServer as a service that starts automatically at system boot time.
- On UNIX, you can have the AdminServer started at system startup by editing your boot script to execute the PROADSV command.

For information on starting the AdminServer, see the [“Starting the AdminServer”](#) section on page 10–16.

2. Create and/or modify Unified Broker configurations using any of the following options:

Note: The properties file that comes installed with your Unified Broker product includes one sample Unified Broker and NameServer instance for each type of Unified Broker that you can use as a guide.

- **Mergeprop utility** — A command-line utility you can use through the **Proenv** command-line interface to manage the contents of all properties files of which the `ubroker.properties` file pertains to the Unified Brokers discussed in this section. The utility supports functionality similar to the Progress Explorer tool. For more information about the mergeprop utility, see the [“Mergeprop utility overview”](#) section on page 10–26.
- **Progress Explorer** — This graphical user interface tool that you can use locally in a Windows machine or remotely from any Windows machine to access configurations installed on UNIX or in Windows machines. See the Progress online help for details about using the Progress Explorer tool to configure Unified Broker properties files.

- **Command-line utilities** — A command-line tool for Windows and UNIX that allows you to control basic activities such as starting, stopping, and querying servers and validating property files associated with OpenEdge products. For more information about the command-line utilities, see [Chapter C, “Command and Utility Reference.”](#)

Note: The properties file that comes installed with your Unified Broker product includes one sample Unified Broker and NameServer instance for each type of Unified Broker that you can use as a guide.

If you plan to configure instances on a UNIX host, you must modify the properties file (`ubroker.properties`) for each Unified Broker instance directly on the host.

Note: To perform most configuration and administrative tasks, use either the mergeprop utility or the Progress Explorer tool because each offers more capabilities than does the command-line utility.

3. Using the Progress Explorer (or the management utility for your Unified Broker product), start up each Unified Broker instance. As it starts, each Unified Broker instance starts additional processes or accesses resources, depending on the product and its configuration.
4. A client can now make a Unified Broker connection request after you ensure that it knows:
 - The correct network location of the NameServer to access.
 - The Application Service name required to connect to the broker that the client needs.

At any time after this step, you can also use any of the appropriate management utilities (mergeprop, Progress Explorer tool, or command-line) to shut down or query the status of any running Unified Broker instance.

5. When you shut down an AdminServer process at any time and—if you have not already shut the Unified Broker instance that it controls —the instance shuts down automatically when you shut down the AdminServer.

During Unified Broker operation, in addition to checking NameServer and Unified Broker status using the Progress Explorer and utilities, you can also review log files being generated by the NameServer and Application Server instance.

The properties file that comes installed with your Unified Broker product includes one sample Unified Broker and NameServer instance for each type of Unified Broker that you can use as a guide.

Understanding and Using the AdminServer

As noted in [Table 10–1](#), the AdminServer is the central control of the Progress Explorer Framework. It facilitates the tasks associated with managing and configuring your installation by ensuring that start and stop requests initiated by OpenEdge products are recognized. In addition to the footnotes identified in [Table 10–1](#), note these points about the AdminServer:

- To start and stop the AdminServer, you can either enter the PROADSV command on the Proenv command line or access the **Services** tab by choosing **Control Panel**→**Administrative Tools**→**Services**.
- To manage and configure plugins such as WebSpeed or AppServer, you can use the Progress Explorer tool.
- Minimize the possibility of a security breach by not starting AdminServer as root.

Minimize the potential for security risks through the AdminServer functionality by ensuring that you do not start the AdminServer as root. Keep in mind that if you do start the AdminServer in this state, all broker processes by default to start as root, leaving your entire system vulnerable to security issues.

- The AdminServer has an extensible framework to host the OpenEdge products as plugins.

The AdminServer loads the plugins and can accept local and remote requests from the Progress Explorer, mergeprop utility, and the command-line utilities. However, the actual work is performed within the plugins themselves as they provide the specific management functions for a particular product.

Starting the AdminServer

In Windows, the AdminServer starts automatically and runs as a Windows Service (AdminService for OpenEdge). From the desktop, you can perform a variety of administrative tasks. On UNIX, the command-line utility, PROADSV, supports a variety of tasks you can perform in support of the AdminServer. This section notes each method.

From the Windows desktop

Use the following procedure to start the AdminServer.



To start the AdminServer on your machine:

- Choose **Start→Programs→Administrative Tools→Services**. Select the **AdminService for OpenEdge 10.1B**, and double-click. The **AdminService for OpenEdge 10.1B Properties** dialog box appears. Choose **Start**, then choose **OK**.

Notes: If **Administrative Tools** is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** check box, then choose **OK**.

If you start the AdminServer, using a specific username and password, that user must have Administrator rights.

For most product installations, the AdminServer is set to Autostart.

Using PROADSV on UNIX

A command-line utility, PROADSV, supports OpenEdge administrative capabilities on UNIX. PROADSV allows you to start up, shut down, and query the status of the AdminServer, among other tasks. See [Appendix C, “Command and Utility Reference”](#) for detailed syntax information.

Stopping the AdminServer

You can stop the AdminServer from the desktop or using the command-line utility, PROADSV. This section notes each method.

Performing the task from the Windows desktop

Use the following procedure to stop the AdminServer from the Windows desktop:



To stop the AdminServer on your machine:

- Choose **Start→Programs→Administrative Tools→Services**. Select the **AdminService for OpenEdge 10.1B**, and double-click. The **AdminService for OpenEdge 10.1B Properties** dialog box appears. Choose **Stop**, then choose **OK**.

Note: If **Administrative Tools** is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** check box, then choose **OK**.

Performing the task using PROADSV on UNIX

A command-line utility, PROADSV, supports OpenEdge administrative capabilities on UNIX. PROADSV allows you to start up, shut down, and query the status of the AdminServer, among other tasks. See [Appendix C, “Command and Utility Reference”](#) for detailed syntax information.

Changing the default port

This section discusses ways to change the AdminServer's default port, 20931.

Performing the task from the Windows desktop

In Windows, the AdminServer runs as a service. The AdminServer is configured to start automatically. However, you can choose to change the listening port for the AdminServer as shown in the following code fragment from the AdminServer rmi registry:

```
[PluginPolicy.Progress.AdminServer]

pluginclasspath={!{value-of:classpath}}
classpath=$DLC/java/...
#In the following code snippet, the port sets the AdminServer rmi registry port
number, the adminport sets the database plugin port, and the jvmargs sets the
database log level to the maximum setting allowed.

port=4321
adminport=7899
```

Note that if you specify values for the `-port` on the command line, these values override values defined in the `%DLC/properties/AdminServerPlugins.properties` file.

Performing the task using PROADSV on UNIX

PROADSV is a command-line utility that you can enter on the `Proenv` command line to support OpenEdge administrative capabilities on UNIX. PROADSV allows you to start up, shut down, and query the status of the AdminServer, among other tasks. See [Appendix C, “Command and Utility Reference”](#) for detailed syntax information.

Also, note that you can also use the code fragment identified in the [“Performing the task from the Windows desktop”](#) section on page 10–19 on UNIX, too.

Changing the startup setting (in Windows only)

For most product installations, the AdminServer is set to Autostart. You can change this setting to Manual mode.

Performing this task from the Windows desktop

Use the following procedure from the Windows desktop:



To change the startup settings from the Windows desktop:

1. In Windows operating systems, choose **Start→Programs→Administrative Tools→Services**. Select the **AdminService for OpenEdge 10.1B** and double-click. The **AdminService for OpenEdge 10.1B Properties** dialog box appears. Choose **Manual** in the **Startup type** field, then choose **OK**.

Note: If **Administrative Tools** is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** check box, then choose **OK**.

2. Modify the [PluginPolicy.Progress.AdminServer] group of the \$DLC/properties/AdminServerplugins.properties.file to use additional command line startup options.

Using PROADSV on UNIX

PROADSV is a command-line utility that you can enter on the Proenv command line to support OpenEdge administrative capabilities on UNIX. PROADSV allows you to start up, shut down, and query the status of the AdminServer, among other tasks. See [Appendix C, “Command and Utility Reference”](#) for detailed syntax information.

Running more than one AdminServer

Keep the following points in mind if you plan to run more than one version of the same OpenEdge software:

- You must run multiple AdminServers. That is, each version requires its own, dedicated AdminServer. For example, if you install and use the OpenEdge 10.1A software and then install a second version, this time of the OpenEdge 10.1B software, each installation requires its own unique AdminServer.

- You can use default port values only for one of your installed versions. Therefore, contention over default values among multiple installation can occur, and must be avoided, since many of the ports that support each version of your OpenEdge software will initially contain default values.

For example, `-port`, `-adminport`, and `agent.properties` file (all of which can be set in the `adminserver.plugins.properties` file, and the `agent.properties` file used only if you are using OpenEdge Management) initially contain default values. It is recommended that you evaluate your port configuration needs **before** running a second, or another, OpenEdge installation in production mode. This pro-active effort helps you to ensure that duplicate ports do not conflict in their attempt to use identical default values. See the [“Performing the task from the Windows desktop”](#) section on page 10–19 for an example of the `adminserver.plugins.properties` file.

Note: The default value available for the `-adminport` is automatically changed for each major OpenEdge release.

Querying the AdminServer

UNIX users generally make more use of the `PROADSV` utility. However, in Windows, you can use the `Proenv` utility with the **Proenv** window, as shown:

Syntax

```
proadsv -query -port 9998
```

`-query`

Displays the AdminServer status.

`-port`

Specifies the listening port for the AdminServer. This is needed if you specified a port other than the default port when you started the AdminServer.

Note: If you specify values for the `-port` or `-adminport` on the command line, these values override values defined in the `%DLC/properties/AdminServerPlugins.properties` file.

Additional AdminServer considerations

Note the following points that pertain to AdminServer usage:

- Before you start a WebSpeed or AppServer application, you must start the AdminServer.
- The AdminServer User-Group Authorization feature requires that you have privileges set to allow you access and operational privileges for the AdminServer. For details to implement this feature on UNIX systems, see the [“How to implement the User-Group Authorization feature”](#) section on page 8–12. For details to implement this feature in Windows platforms, see the Windows online help topic “Establishing AdminServer Authorization Options during the Installation.”

For other details about the AdminServer, Windows users should refer to the [“Getting started with the AdminServer”](#) section on page 7–11 and UNIX users should refer to the [“Getting started with the AdminServer”](#) section on page 8–10.

AdminServer-related authorization option

In Windows, you can optionally establish AdminServer authorization options for OpenEdge products that support the AdminServer during the installation process. These options are:

- **User Authorization** — To require each individual user to provide a valid user name and password before the AdminServer can be started.
- **Group Authorization** — To setup user-defined group names for which operational privileges, at a group level, are required. Group name definitions must conform to specific guidelines.

For information about these options, see the following Windows online help topics: “Establishing AdminServer Authorization Options during the Installation,” and “User-defined Group Name Conventions and Restrictions.”

AdminServer Logging

There are logging entries that are specifically related to user authentication and authorization.

Using the Progress Explorer tool

Progress Explorer is a graphical administration tool that provides an easy way to manage OpenEdge servers. It runs as a Windows client of the AdminServer, but can also be used with UNIX installations, depending on how you have your machines set up.

OpenEdge Servers that Progress Explorer supports

The Progress Explorer manages the following OpenEdge servers:

- AppServer.
- AppServer Internet Adapter (AIA).
- Database.
- DataServer for MS SQL Server.
- DataServer for ODBC (Windows only).
- DataServer for ORACLE.
- NameServer.
- OpenEdge Adapter for SonicMQ.
- WebSpeed Adapter.
- WebSpeed Messenger.
- WebSpeed Transaction Server.

Progress Explorer tool capabilities

Using the Progress Explorer tool, you can:

- Create new instances of OpenEdge servers and configure their property settings.
- Modify property settings of existing OpenEdge server instances.
- Start and stop OpenEdge servers.

- Monitor the status of OpenEdge servers.
- Remove existing OpenEdge product instances.



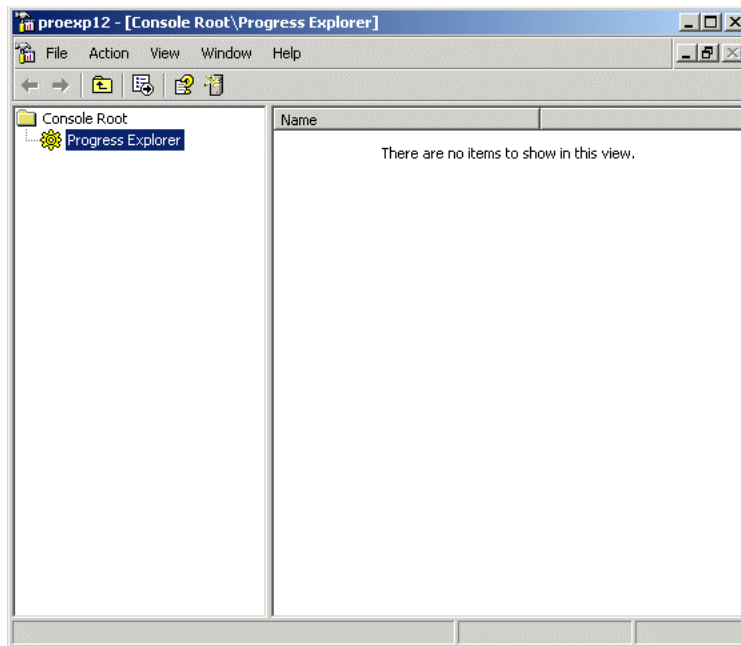
To use the Progress Explorer configuration tools, you must first start the Progress Explorer tool and connect to the port where the AdminServer is running. In this situation, the AdminServer must be running on the same host where the OpenEdge products you want to start, stop, or query are installed.

When you establish a connection to the AdminServer, Progress Explorer presents a tree view of all the OpenEdge products to which the AdminServer grants you administrative access. For complete instructions on using Progress Explorer, see the Progress Explorer online help.



To launch the Progress Explorer:

1. From the **Start** menu choose **OpenEdge→Progress Explorer Tool**, or from the OpenEdge Program Group, double-click the Progress Explorer Tool icon. When you start Progress Explorer, the main window appears:



2. Follow the instructions in the Progress Explorer online help to establish a connection to an AdminServer.

Saving configurations

Progress Explorer allows you to save:

- NameServer, AppServer, AppServer Internet Adapter, OpenEdge Adapter for SonicMQ, WebSpeed Transaction Server, WebSpeed Messenger, DataServer for MS SQL Server, DataServer for Oracle, and DataServer for ODBC configurations.

Configurations you create with any other Progress Explorer configuration tool are saved in the `ubroker.properties` file. You can edit the `ubroker.properties` directly with any text editor. Progress Explorer will automatically update if the `ubroker.properties` is manually edited. When you want to edit the configuration of a NameServer, AppServer, AppServer Internet Adapter, OpenEdge Adapter for SonicMQ, WebSpeed Transaction Server, or WebSpeed Messenger, use only one configuration tool at a time.

- Database configurations

Database configurations you create with the Progress Explorer Database configuration tool are saved in the `conmgr.properties` file. Do not edit the `conmgr.properties` file directly; use Progress Explorer to create and edit database configurations.

- NameServer, AppServer, AppServer Internet Adapter, OpenEdge Adapter for SonicMQ, WebSpeed Transaction Server, WebSpeed Messenger, DataServer for MS SQL Server, DataServer for Oracle, and DataServer for ODBC configurations.

Configurations you create with any other Progress Explorer configuration tool are saved in the `ubroker.properties` file. You can edit the `ubroker.properties` directly with any text editor. Progress Explorer will automatically update if the `ubroker.properties` is manually edited. When you want to edit the configuration of a NameServer, AppServer, AppServer Internet Adapter, OpenEdge Adapter for SonicMQ, WebSpeed Transaction Server, or WebSpeed Messenger, use only one configuration tool at a time.

Mergeprop utility overview

The mergeprop utility allows you to manage the content of OpenEdge property files. As previously mentioned in this chapter, property files store configuration information that specifies and controls the behavior of various components.

The mergeprop utility is an alternative, fully supported tool by which you can update the `comgr.properties` file, `ubroker.properties` file, and other property files when either the Progress Explorer tool is not available or you choose to use this approach. Presented through a Proenv command-line interface, it provides a consistent means to manage and maintain property files, by direct user action or programmatically. It also implements several additional value types not previously supported in property files for increased flexibility and ease of maintenance.

When using the mergeprop utility, the user or program specifies an optional *target file* in which the existing property file is to be modified and a *delta file* which contains the changes to be made.

Detailed instructions are presented in the [“Using the mergeprop utility”](#) section on page 10–27.

Property value

[Table 10–3](#) identifies the value types you can use with the mergeprop utility to manage property files. These property files allow you to:

- Use the value of another existing property by reference.
- Specify a list of any number of syntactically valid values. The list entries are evaluated sequentially, and the first to be successfully resolved is the value of the property.
- Specify a hexadecimal value.

In addition, two special value types can be included in the delta file specified with the mergeprop utility. (See the [“Delta file”](#) section on page 10–30.) These value types are valid **only** in the context of a delta file:

- A value formed by appending a specified string to a current value.
- A system-generated globally unique identifier (UUID, or universally unique ID).

Using the mergeprop utility

This section provides the syntax and instructions to use the mergeprop utility from a command line.

Command syntax

The following table presents the mergeprop syntax:

Operating system	Syntax ¹
UNIX Windows	<code>mergeprop -type <i>file_type</i></code> <code>[-action <i>operation_type</i> [<i>group_name</i>]]</code> <code>[-target <i>target_file</i>]</code> <code>[-delta <i>delta_file</i>]</code> <code>[-validate]</code> <code>[-nobackup]</code> <code>[-silent]</code> <code>[-recurse]</code>

1. Command switches can be defined in any order following the mergeprop command.

Refer to [Table 10–3](#) for the details about valid values for argument variables.

Command switches and arguments

Table 10–3 summarizes the syntax elements used with the mergeprop command. Refer to the “Mergeprop utility overview” section on page 10–26 for additional descriptive information.

Table 10–3: Command line input to the mergeprop command (1 of 2)

Switch ¹	Arguments	Notes
-type (required)	ubroker database tools plugin none	Each argument (other than none) implies a specific target file in the properties directory. See the “File type” section on page 10–30.
-action	update (default) create delete list <i>group_name</i> listall <i>group_name</i>	If no action is specified, update is assumed by default. The list and listall actions require an additional argument, the name of the property group to be displayed (for example, ubroker.AS.asbroker1). Do not include the square brackets ([]) that enclose the group name in the ubroker.properties file. On update and create actions, groups listed with no properties in the delta file are ignored.
-target (optional)	Path to the property file to be modified.	If you are updating a property file that is in the <i>OpenEdge-install-dir/properties</i> subdirectory, you can omit this option. Only use this option when the property file you plan to update exists in a location other than the <i>OpenEdge-install-dir/properties/</i> subdirectory.
-delta	Path to the delta file containing changes to be made.	File containing create, update, or delete changes.
-validate	None.	Performs all processing without actually making changes to the target file. This option lets you test for errors.

Table 10–3: Command line input to the mergeprop command (2 of 2)

Switch ¹	Arguments	Notes
-nobackup	None.	Does not create a backup to the target file before making changes. Unless you invoke this option, mergeprop saves a copy of the original target file in the same directory. The backup copy has a system-generated unique string appended to the name (for example, <code>ubroker.properties_31420040644533</code>).
-silent	None.	Suppresses all messages.
-recurse	None.	Lists or deletes all groups, server groups, and configurations associated with the specified database. The recurse option is only valid when the file type is specified as database and the action is identified as either <code>list</code> or <code>delete</code> .

1. Command switches can occur in any order following the mergeprop command.

Target file

The *target file* is the existing property file on which you are operating. You can add, delete, modify or list properties in the target file. The mergeprop program automatically creates a backup of the original target file, unless you instruct it not to do so. You can also list existing properties without making any changes.

You can explicitly specify a target file, but it is not necessary to do so if you are operating on one of the standard property files listed in [Table 10–4](#). The file type that you provide as input implies a specific property file, which the program targets by default if no file is specified. These standard property files are located in the *OpenEdge-Install-Directory*\properties directory.

Table 10–4: Property files managed by the mergeprop utility

Property file	Components configured	Corresponding file type
ubroker.properties	Unified Broker products, including all products managed through the Progress Explorer tool with the exception the database products.	ubroker
conmgr.properties	Database startup parameters.	database
JavaTools.properties	Client-side tool configuration, for example the Progress Explorer and command line tools.	tools
AdminServerPlugins.properties	Plugin products loaded by the AdminServer.	plugins

If explicitly specified, the target file is expressed as an argument to the `-target` command switch.

Delta file

To use the utility to make changes, you must list the affected groups and properties in a delta file. The delta file must specify at least one property group; it can also specify one or more properties and associated values. The content of the delta file must conform to the syntax rules for property files, as explained in the [“Logical structure and syntax of property files”](#) section on page 10–35.

Note: When simply listing (not changing) properties, you do not specify a delta file.

The delta file is expressed as an argument to the `-delta` command switch.

File type

There are five distinct property file types:

- Ubroker
- Database

- Tools
- Plugin
- None

As indicated in [Table 10–4](#), one standard property file of each type is found in the *OpenEdge-install-dir\properties* directory.

Specifying the file type enables the mergeprop utility to process delta and target files appropriately. It also makes it unnecessary to explicitly identify the target file, unless you are operating on a copy or test file other than the standard file in the *properties* directory. The program can recognize “none” as a valid type and perform default processing, but you should provide a specific type as input.

The file type is expressed as an argument to the `-type` command switch.

Action switch argument values

The mergeprop utility operates on the target file, based on action you provide as input, in one of the following ways:

- **Update** — Creates any new property groups and modifies any existing groups found in the delta file. Properties in the target file are updated to match those in the delta file. This operation is performed by default if you do not explicitly specify an action.
- **Create** — Creates new property groups listed in the delta file, with properties as specified in the delta file. (The delta file must contain only new groups; inclusion of a group that already exists in the target file causes an error and cancels the operation.)
- **Delete** — Removes from the target file any property groups listed in the delta file. The entire groups are deleted; individual properties are not processed. No exception occurs if the delta file contains groups that do not exist in the target file; such groups are simply ignored.
- **List** — Displays (to `stdout`) all properties and values defined specifically for a given group. Inherited properties are not displayed.
- **List all** — Displays (to `stdout`) all properties and values defined for a given group, including inherited properties. (This action applies only to the database.)

In this context, group refers to a group as listed in the `ubroker.properties` file. For example, `[UBroker.AS.asbroker1]`. (Note that the brackets are not part of the command.)

For more information about groups, see the [“Logical structure and syntax of property files”](#) section on page 10–35.

Note: The **List** and **List all** actions are useful for creating a delta file. You can redirect output to a file and use the result as a template for modifying existing instances or creating new ones.

The following examples demonstrate how you can perform various modifications using the mergeprop utility.

The following example has two parts. The first code fragment shows the contents of the delta file in which a new AppServer Broker instance, addasbroker2, is defined. The contents of this delta file is based on minor changes made to the sample default broker asbroker1, as shown:

```
$ cat addasbroker2
[UBroker.AS.asbroker2]
appserviceNameList=asbroker2
brokerLogFile=@{WorkPath}\asbroker2.broker.log
portNumber=3092
uuid=932.99.999.XX:1ee77e:cf3bbe3d33:-8000
```

The command in the following example adds the new asbroker2 to the standard, OpenEdge-supplied ubroker.properties file:

```
$ mergeprop -type ubroker -action update -delta addasbroker2
```

This same command structure can be used to update a group.

Note: On an add action, you are only required to specify those properties whose values you intend to override. Default values are applied in all other circumstances.

The second example demonstrates how to add and delete property specified as an “environment” property to the `asbroker2` previously created. It is also defined in two parts. The following code fragment shows the environment property is being added to the `asbroker2` definition in the `ubroker.properties` file:

```
$ cat asbroker2prop
[UBroker.AS.asbroker2]
environment=asbroker2

[Environment.asbroker2]
MYENV=hello
```

```
mergeprop -type ubroker -action update -delta asbroker2prop
```

It also is helpful to know how to perform a deletion. In reviewing the third example, keep in mind that you can only perform group-level deletions; you cannot delete a single property within a group. The following example shows how to delete the instance of `asbroker2`:

```
$ mergeprop -type ubroker -action delete ubroker.AS.asbroker2
```

The fourth example lists the properties defined specifically for the `UBroker.AS.asbroker1` group in `ubroker.properties`, omitting inherited properties, as follows:

```
mergeprop -type ubroker -action list UBroker.AS.asbroker1
```

The fifth example lists all properties of the `UBroker.AS.asbroker1` group, including inherited properties, as shown:

```
mergeprop -type ubroker -action listall UBroker.AS.asbroker1
```

The sixth example lists all properties, including inherited properties, of the `FMCONFIGCLI.OSFI` group in the file `installation-path\%DLC%\properties\JavaTools.properties`, as shown:

```
$ mergeprop -type tools -action listall FMCONFIGCLI.OSFI
```

As shown in the seventh example, it is helpful to know how to list a full group definition. The following code fragment shows how to enter the mergeprop command for a full database group definition. In the following example, the sports database, is referenced and its full group definition which lists all configurations and server groups associated with the sports database noted:

```
$ mergeprop -type database -action listall sports -recurse
[database.sports]
Autostart=false
Configurations=sports.defaultconfiguration
DatabaseName=/usr1/sports
DefaultConfiguration=sporsts.defaultconfiguration
DisplayName=sports
MonitoredLicense=true
[configuration.sports.defaultconfiguration]
AfterImageProcess=false
AsynchronousPageWriters=1
BeforeImageProcess=true
Database=sports
DisplayName=defaultConfiguration
Monitored=true
OtherArgs=
ServerGroups=sports.default.configuration.defaultservergroup
WatchDogProcess=true
[servergroup.serverGroups=sports.default.configuration.defaultservergroup]
Configuration=sports.defaultconfiguration
DisplayName=defaultServerGroup
Port=4441
Type=both
```

Referencing the previous code fragment, you could update a port specification for the sports database using the following example commands:

```
$ cat changeport
[servergroup.sports.defaultconfiguration.defaultservergroup]
port=4444
```

```
$mergeprop -type database -action update -delta changeport
```

Logical structure and syntax of property files

All property files use a hierarchical structure consisting of named groups and subgroups. Each group or subgroup can define a set of properties, for which the values can be either specified or null.

A subgroup inherits all properties from its parent group. By default, it also inherits the values of those properties. Within a subgroup, inherited defaults can be overridden and additional properties can be defined. The lowest level subgroup defines a specific instance of the component type.

Note: Properties are valid only if they are allowed by the appropriate schema file. An attempt to create an unsupported property results in an error.

The syntax in a property file is as follows:

- **Group names** — Names are enclosed in square brackets. A subgroup name consists of the parent group's name followed by a period and the identifier for the subgroup. Thus, the names [WebSpeed], [WebSpeed.Messengers], and [WebSpeed.Messengers.CGIIP] form a three-level hierarchy of property groups.
- **Properties and values** — Property name-value pairs are listed immediately following the name of the group for which they are defined. The property name is followed by an “equals” sign and, optionally, a value. For example, `controllingNameServer=` defines a property with a null value; `controllingNameServer=NS1` assigns a specific value to that property.

For example, consider the following groups defined in the as-installed version of `ubroker.properties`:

```
[UBroker]
    controllingNameServer=
    svrLogEntryTypes=
[UBroker.AS]
    svrLogEntryTypes=ASPlumbing,DB.Connects
    description=AppServer Broker
[UBroker.AS.asbroker1]
    controllingNameServer=NS1
    description=A sample AppServer setup for State-reset
```

The top-level [UBroker] group defines a set of properties that are inherited by the subgroup [UBroker.AS] and by all other subgroups.

The subgroup [UBroker.AS] defines properties for all AppServer instances. It assigns a default value to the inherited `srvrLogEntryTypes` property, and it defines an additional description property.

The subgroup [UBroker.AS.asbroker1] defines an AppServer instance. It assigns a value to the `controllingNameServer` property inherited from [UBroker], and it overrides the value of the description property inherited from [UBroker.AS].

Property value formats

This section provides a summary of the supported formats for expressing property values. These formats are presented in three categories:

- Newly supported formats (introduced in OpenEdge 10) that are valid in all property files.
- Formats that are valid only in delta files used as input to the mergeprop utility.
- Formats that were supported prior to OpenEdge 10.

Table 10–5 lists the formats that were introduced in OpenEdge 10 for use in all property files.

Table 10–5: New value formats supported in all property files

Description	Syntax and example
Reference to another property value.	<code>!{value-of:group.property}</code> Example: <code>jvmargs=!{value-of:Common.jvmargs}</code>
Reference to a Java system property.	<code>!{SystemProperty:java_property}</code> Example: <code>userName=!{SystemProperty:userName}</code>
List of references to be evaluated sequentially. The first reference to be resolved is used. The last entry can be an explicit value. The delimiter between references is a question mark (?), and the list must also be terminated with a question mark.	<code>?value1?value2?value3...value-n?</code> Example: <code>description=?!{SystemProperty:userName}?!{value-of:NameServer.NS1.hostName}?NS1 Host?</code>
Hexadecimal value.	<code>hex_value</code> Example: <code>srvrLoggingLevel=0x0BF</code>

Table 10–6 lists the formats that were introduced in OpenEdge 10 for use exclusively in delta files used as input to the mergeprop utility.

Table 10–6: New value formats supported in mergeprop delta files only

Description	Syntax and example
Value formed by appending a specified string to the existing value.	<code>!{current-value}append_string</code> Example: <code>description=!{current-value} UPDATED</code>
Reference to a Java system property.	<code>!{newValue:UUID}</code> Example: <code>uuid=!{newValue:UUID}</code>

Table 10–7 lists the remaining supported formats, which were introduced prior to the release of OpenEdge 10.

Table 10–7: Value formats supported prior to OpenEdge 10

Description	Syntax and example
An explicit integer or string constant.	<code>value</code> Example: <code>portNumber=3095</code>
Reference to a system environment variable.	<code>\${env_variable}</code> Example: <code>workDir=\${WORKDIR}</code>
Reference to a Windows registry value.	<code>@{registry_value}</code> Example: <code>workDir=@{WorkPath}</code>

Ubroker.properties file and product configurations

The `ubroker.properties` file stores all the configuration definitions for each instance of the following OpenEdge products:

- OpenEdge NameServer.
- AppServer.
- AppServer Internet Adapter.
- OpenEdge Adapter for SonicMQ.
- DataServer for MS SQL Server.
- ODBC DataServers (in Windows only).
- ORACLE DataServer.
- WebSpeed Transaction Server.

The UNIX and Windows `ubroker.properties` files are the same except for platform-specific differences (for example, differences in directory path separators and the differences between environment variable references on UNIX and registry references in Windows).

There is one copy of this file local to each OpenEdge installation. The AdminServer reads and updates the file according to your instructions using the Progress Explorer and management utilities. The `ubroker.properties` file is installed in the `properties` subdirectory of the OpenEdge installation directory (for example, `$DLC/properties/ubroker.properties` on UNIX, or `%DLC%\properties\ubroker.properties` in Windows). In order for the AdminServer to access the `properties` file, the file must reside in this directory.

Unified Broker products and associated clients

Table 10–8 identifies each Unified Broker product and indicates the types of clients that can use the Unified Brokers’ services.

Table 10–8: Unified Broker products and the clients that they support

This Unified Broker product . . .	Supports these types of clients . . .
AppServer	ABL clients (including other AppServers and WebSpeed instances) and Open Clients.
AppServer Internet Adapter	ABL clients (including AppServer and WebSpeed instances).
DataServers	ABL clients (including AppServer and WebSpeed instances).
OpenEdge Adapter for SonicMQ	ABL clients (including AppServer and WebSpeed instances).
WebSpeed	The WebSpeed Messenger, which directs Web client requests to WebSpeed Transaction Servers.

Of these clients, you can use the Unified Broker administration framework to manage only WebSpeed Messengers. For specific information on configuring these clients, see your Unified Broker product documentation.

Unified Broker installation prerequisites

Before you install a new Unified Broker version, either to overwrite an existing installation or to add additional OpenEdge components to the current installation, make copies of your `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences` files and place them in another directory. This is necessary because the new installation **automatically** upgrades the files in the `install-path\properties` directory. After you have finished your new installation, replace the newly installed versions of these files with your copies. When you start the AdminServer, your older files will be updated to match the current standards for these files.

When you uninstall an existing Progress or OpenEdge product, the process copies `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences`, the three files in the `OpenEdge-install-path\properties` directory, to `%TEMP%`. After installing a new OpenEdge Release 10.1B product, you can manually copy back the files from `%TEMP%`.

Under certain conditions, you might have to modify this file. You can use the mergeprop utility or a text editor.

Note: Each configuration definition contains environment variables, registry entries, and property settings for each product instance. Progress Explorer and the associated command-line configuration utilities use this file to store and validate the configurations for the products.

Ubroker.properties file structure

The `ubroker.properties` file consists of a hierarchical structure of configuration entities, where 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. The AppServer configurations in `ubroker.properties` are shown in the following table:

Configuration entity name	Configuration entity function
[UBroker]	Defines default property settings for all NameServer, AppServer, DataServer, and WebSpeed Transaction Server 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> .

Note: Each configuration definition contains environment variables, registry entries, and property settings for each product instance. Progress Explorer and the associated command-line configuration utilities use this file to store and validate the configurations for the products.

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 the redefinition 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. The `conmgr.properties` file is installed in the `OpenEdge-install-dir\properties\conmgr.properties`.

AdminServer and requirements to set an owner for the Broker

The AdminServer honors a user's permissions, according to the user's profile that was used to start an AdminServer. For example, a user who intends to start an AdminServer for another user's process must have the rights to start this second process. These rights or settings must be previously set in the broker's Owner Information properties category. For more information about the Owner Information and the owner feature, see the Progress Explorer online help.

Working with the supported products

For the definitions and usage of all properties supported in the properties file, see the comments in the `ubroker.properties.readme` file that is available at

`C:\OpenEdge-install-dir\properties` from the installed `ubroker.properties` file. For more information on:

- Each configuration management and validation utility, see *OpenEdge Application Server: Administration* and *OpenEdge Data Management: Database Administration*.
- Maintaining `ubroker.properties` for AppServer installations, see *OpenEdge Application Server: Administration*.
- Maintaining `ubroker.properties` for DataServer installations, see the OpenEdge DataServer Guide for your DataServer product.
- Maintaining `ubroker.properties` for WebSpeed installations, see *OpenEdge Application Server: Administration*.
- Maintaining `ubroker.properties` for NameServer installations, see the [“Ubroker.properties file and product configurations”](#) section on page 10–38. Also see the specific OpenEdge product manual, referencing the section that includes the NameServer in its configuration.
- Maintaining `ubroker.properties` for AppServer Internet Adapter installations, see *OpenEdge Application Server: Administration*.
- Maintaining `ubroker.properties` for the OpenEdge Adapter for SonicMQ installations, see *OpenEdge Application Server: Administration*.

Editing and validating the content of the `ubroker.properties` file

Progress Explorer can be used in Windows platforms, and can connect remotely to UNIX host machines to perform configuration activities. Use the `mergeprop` utility on either platform if the Progress Explorer is not available.

To ensure proper run-time processing, the file must be named `ubroker.properties` and must be located in the `properties` subdirectory of the OpenEdge installation directory.

Guidelines for editing the `properties` file

In general, you should update all configurations (local or remote) using either the Progress Explorer tool or `mergeprop` utility. If you must update a configuration locally using a text editor:

- Do not directly change the values in the `ubroker.properties` file unless you have a complete understanding of how the changes affect Unified Broker components.
- Make a copy of this file, edit the copy, and verify the result before replacing the original with your edited copy.
- For complete definitions of all the properties and detailed information on how to set them, see the comments included in the `properties` file.
- Verify the result using the appropriate configuration validation utilities.



To edit the file and create or modify a product configuration:

1. Make a backup copy from the installed `ubroker.properties` file (and name it, for example, `test.properties`).
2. Make changes to `test.properties` with a text editor.
3. Run the appropriate validation utility on `test.properties` using the `-propfile` option to validate your changes. For a complete list of the command-line utilities you can use to validate property file changes, see [Table 10–10](#).

Shut down any running brokers or NameServers using the `-stop` option of the appropriate configuration management utilities (for example, `nsman` and `asbman`).

4. Copy `test.properties` to `ubroker.properties` in the `properties` subdirectory of the OpenEdge installation directory. If you might return to the previous configuration, store a backup copy of the `ubroker.properties` file before replacing it with your modified version.

5. Restart your brokers and NameServers using the `-start` option of the appropriate configuration management utilities. For a complete list of the command-line utilities you can use to configure property files, see [Table 10–9](#).

For more information on editing and validating the `ubroker.properties` file to configure a NameServer, see the “[Editing and validating the content of the `ubroker.properties` file](#)” section on page 10–42. For more information on editing and validating the file for each Unified Broker product, see your product documentation.

Command-line utilities reference

The command-line management utilities are a set of utilities for Windows and UNIX that allow you to manage existing configurations. Like the Progress Explorer tool, the command-line management utilities run as clients of the OpenEdge AdminServer to manage the NameServer and Unified Broker products.

Using these utilities, you can locally or remotely start, stop, manage, and monitor the status of Unified Broker execution. Unlike the Progress Explorer, they do not help you create, remove, or modify properties for Unified Broker configurations.

The framework supports several different product-specific command-line configuration utilities that you can use to manage—that is, start, stop, and query activities—installed OpenEdge server products.

[Table 10–9](#) identifies the product-specific command-line utilities available.

Table 10–9: Command-line utilities to start and stop installed OpenEdge products

(1 of 2)

To start and stop . . .	Use this utility . . .
A configured OpenEdge Adapter for SonicMQ.	ADAPTMAN
A configured AppServer.	ASBMAN
The current configuration of an OpenEdge database, or its agent.	DBMAN
A configured DataServer for Microsoft SQL Server.	MSSMAN
A configured NameServer.	NSMAN

Table 10–9: Command-line utilities to start and stop installed OpenEdge products*(2 of 2)*

To start and stop . . .	Use this utility . . .
The operation of a configured DataServer for ODBC.	ODBMAN
The operation of a configured DataServer for ORACLE.	ORAMAN
And configure the Web Services Adapter.	WSAMAN
The operation of a configured WebSpeed Transaction Server.	WTBMAN

Command-line configuration utilities to validate property files associated with OpenEdge products

OpenEdge supports two approaches to validate property files associated with installed OpenEdge products:

- Mergeprop utility. For more information, see the [“Mergeprop utility overview”](#) section on page 10–26.
- Through command-line utilities that are available to validate property files associated with installed OpenEdge products. [Table 10–10](#) identifies these utilities.

Table 10–10: Command-line utilities to validate property files*(1 of 2)*

To validate property files associated with . . .	Use this utility . . .
An existing OpenEdge Adapter for SonicMQ configuration.	ADAPTCONFIG
An existing AppServer Internet Adapter configuration.	AIACONFIG
An existing configuration for an AppServer.	ASCONFIG
An existing configured DataServer for Microsoft MS SQL.	MSSCONFIG

Table 10–10: Command-line utilities to validate property files *(2 of 2)*

To validate property files associated with . . .	Use this utility . . .
An existing configured NameServer.	NSCONFIG
An existing configured DataServer for ODBC.	ODBCCONFIG
An existing configured DataServer for ORACLE.	ORACONFIG
The configured Web Services Adapter.	WSACONFIG
An existing configured WebSpeed Messenger.	WSCONFIG
All databases.	DBCONFIG

For more information on starting and managing the NameServer using the Progress Explorer and NSMAN utility, see [Chapter C, “Command and Utility Reference,”](#) and [Chapter E, “NameServer and Name Server Load Balancing Details.”](#)

Starting and Running OpenEdge

This chapter describes how to start up and connect to an OpenEdge database, as detailed in the following sections:

- [Starting OpenEdge in Windows platforms](#)
- [Starting OpenEdge on UNIX platforms](#)
- [Running OpenEdge clients and servers on a network](#)

Starting OpenEdge in Windows platforms

OpenEdge startup commands differ with certain operating systems, user interfaces, and network software. In Windows, you can use the Progress Explorer to start a server. See the [“Configuration”](#) section on page 10–1 for information on launching the Progress Explorer. See the Progress Explorer online help for instructions on connecting to an AdminServer.

If you are an international customer, you can set code pages for different application components at startup. You can also set numerical and date/time formats at startup by specifying internationalization parameters. See [OpenEdge Development: Internationalizing Applications](#) and [OpenEdge Data Management: Database Administration](#) for more information on using internationalization parameters at startup.

Startup and shutdown

You can use either the Client or the Progress Explorer to perform many of the startup and shutdown tasks. These methods provide you with a GUI interface for managing and configuring your databases and servers. If you are not using a Windows environment, or if you prefer a command-line interface, you can choose to enter commands at the command line to perform these tasks. The following sections explain how to use the GUI and command-line interfaces to perform startup and shutdown tasks.

Using the GUI interface

You can use either the Client or Progress Explorer to perform startup and shutdown tasks, as indicated in [Table 11–1](#). To perform one of the tasks listed using the Client, open the properties of the Client and modify the shortcut target as indicated.

To perform one of the tasks using the Progress Explorer, start the Progress Explorer, select the server you want to start or stop, and follow the instructions in the online help.

Note: These instructions refer to the ABL Client. To perform SQL tasks, you must start the SQL Explorer and use the SQL Client. See the SQL Explorer online help for more information about using the SQL Client.

Table 11–1 summarizes tasks and methods to perform startup and shutdown tasks using the graphical user interface (GUI).

Table 11–1: Windows GUI startup and shutdown commands

Task	OpenEdge program group icon	Action
Start the Procedure Editor and connect to a single-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe pathname\db-name -1</i>
Start the Procedure Editor and connect to a multi-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe pathname\db-name</i>
Start the ADE Desktop and connect to a single-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe -p _desk.p pathname\db-name -1</i>
Start the ADE Desktop and connect to a multi-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe -p _desk.p pathname\db-name</i>
Start an OpenEdge batch session and connect to a single-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe -b pathname\db-name -1 -p procedure</i>
Start an OpenEdge batch session and connect to a multi-user database.	Client	Modify shortcut target properties: <i>install-path\bin\prowin32.exe -b pathname\db-name -p procedure</i>
Start a server or broker for an OpenEdge database.	Progress Explorer	See online help. Command-line alternative: <i>proserve pathname\db-name</i>
Shut down a server or broker for an OpenEdge database.	Progress Explorer	See online help. Command-line alternative: <i>proshut pathname\db-name</i>

Using the command-line interface

Startup commands start a Progress session and connect you to a database. [Table 11–2](#) summarize the startup and shutdown commands for the Windows operating system and its functions. For detailed information on these commands and their parameters, see the descriptions of the commands in either *OpenEdge Deployment: Startup Command and Parameter Reference* or *OpenEdge Data Management: Database Administration*.

[Table 11–2](#) summarizes the tasks to perform and the related startup and shutdown commands to use in Windows systems.

Table 11–2: Windows startup and shutdown commands

(1 of 2)

Task	Command
Start a Windows character Procedure Editor and connect to a single-user database.	<code>pro db-name</code>
Start a Windows character Procedure Editor and connect to a multi-user database.	<code>mpro db-name</code>
Start a Windows character client session in batch mode and connect to a single-user database.	<code>bpro db-name -p procedure-name</code>
Start a Windows character client session in batch mode and connect to a multi-user database.	<code>mbpro db-name -p procedure-name</code>
Start an OpenEdge server-group.	<code>proserve -servergroup server-group-name</code>
Start a server or broker for a multi-user OpenEdge database.	<code>proserve db-name -S service-name -H host-name -N network-type</code>
Shut down a multi-user server or broker for an OpenEdge database.	<code>proshut db-name</code>
Start a remote OpenEdge DataServer broker.	<code>probrkr -S service-name -H host-name -N network-type</code>
Start an asynchronous page writer (APW) for a database ¹ .	<code>proapw db-name</code>

Table 11–2: Windows startup and shutdown commands

(2 of 2)

Task	Command
Start a before-image writer (BIW). ¹	<code>probiw db-name</code>
Start an after-image writer (AIW). ¹	<code>proaiw db-name</code>
Start the OpenEdge Watchdog utility. ¹	<code>prowdog db-name</code>
Shut down a remote OpenEdge DataServer.	<code>proshut db-name</code> <code>-S service-name</code> <code>-H host-name</code> <code>-N network-type</code>
Shut down an APW, AIW, BIW, or Watchdog process. ¹	<code>proshut db-name</code> Choose option 1 (Disconnect a User) to disconnect the process.

1. Option available only on Enterprise product.

Starting OpenEdge as a Windows service

To run OpenEdge as a Windows service, you must start ProService before starting a Progress session. To accomplish this, start the Progress Explorer and connect to an AdminServer.

Note: ProService is run as a Windows service. This means it runs under the system account. It does not run under the account the user is currently logged into. You must grant system access to the directory containing the database for ProService to work properly.

Using the Progress Explorer to connect to the AdminServer

You use the Progress Explorer to connect to the AdminServer.



To connect to the AdminServer:

1. From the **Start** menu choose **OpenEdge→Progress Explorer Tool**, or from the Progress Program Group, double-click the Progress Explorer Tool icon. The Progress Explorer Tool icon looks like this:



2. Start the Progress Explorer and establish a connection to an AdminServer. See [Chapter 10, “Configuration,”](#) for more information about Progress Explorer, and see the Progress Explorer online help for detailed instructions on connecting to an AdminServer.

Starting single-user OpenEdge in interactive mode

To start single-user OpenEdge, enter the following command:

```
prowin32 [ db-name ] -1 [ parameters ]
```

db-name

Specifies the database you want to start. (-db is implicit but can be specified)

parameters

Specifies the startup parameters you want to use to describe system and application characteristics. For a detailed description of the Progress startup parameters, see [OpenEdge Deployment: Startup Command and Parameter Reference](#), [OpenEdge Data Management: Database Administration](#) and the Progress DataServer Guides as noted in the following list:

- [OpenEdge Data Management: DataServer for Microsoft SQL Server](#)
- [OpenEdge Data Management: DataServer for ODBC](#)
- [OpenEdge Data Management: DataServer for ORACLE](#)

Starting single-user OpenEdge in batch or background mode

Batch or background processing is convenient for large-scale database updates or procedures that you can run unattended (at night, for example).

To start single-user OpenEdge in batch or background mode, enter the following command:

```
prowin32 [ db-name ] -l -b -p procedure-name [ parameters ]
```

db-name

Specifies the database you want to start.

-b

Specifies that OpenEdge should run in batch mode.

-p procedure-name

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

output-file

Specifies the name of the file that receives all output to the default stream.

Starting the multi-user server or broker

Before you can run multi-user Progress, you must start the multi-user server process. The server process coordinates all the database requests from all the users using a single database. You can use the Progress Explorer to start the multi-user server process, or you can use the command-line interface. The following sections describe these methods of starting the multi-user server process.

Using the Progress Explorer to start the multi-user server process

You can use the Progress Explorer to start a multi-user server process.

Start the Progress Explorer and establish a connection to one or more AdminServers. See [Chapter 10, “Configuration,”](#) for more information about starting Progress Explorer, and see the Progress Explorer online help for detailed instructions on using Progress Explorer to start a multi-user server process.

Using the command line interface to start the multi-user server process

Enter the following command at the command line to start the multi-user server process:

```
proserve db-name [ parameters ]
```

db-name

Specifies the database you want to start Progress against. (-db is implicit)

parameters

Specifies the startup parameters for the broker/server. For a list of broker/server startup parameters, see [OpenEdge Deployment: Startup Command and Parameter Reference](#) and the Progress DataServer Guides as noted in the following list:

- [OpenEdge Data Management: DataServer for Microsoft SQL Server](#)
- [OpenEdge Data Management: DataServer for ODBC](#)
- [OpenEdge Data Management: DataServer for ORACLE](#)

Starting the multi-user server or broker as a Windows service

Before you can run multi-user OpenEdge as a Windows service, you must create an entry in the registry to enable OpenEdge to run as a Windows service. Use the Progress Explorer to create an entry in the registry. The following sections describe these methods.

Using the Progress Explorer to start the multi-user server or broker

Start the Progress Explorer and establish a connection to one or more AdminServers. See [Chapter 10, “Configuration,”](#) for more information about starting Progress Explorer, and see the Progress Explorer online help for detailed instructions on using Progress Explorer to start a multi-user server process.

Starting OpenEdge on UNIX platforms

OpenEdge startup commands differ with certain operating systems, user interfaces, and network software. UNIX provides a series of scripts to run the OpenEdge executables, such as `proserve` to start broker/servers and `mpro` to start multi-user interactive clients. These scripts are tailored for your particular software environment. For information on the script executed by each command, see the description of the command in [Table 11–3](#).

It is important that you observe the following conventions when you enter a command:

- Use lowercase characters for commands on UNIX.
- Enter parameters on UNIX exactly as shown in the syntax descriptions.
- Values can be case sensitive on UNIX. (For example, names of UNIX files are case sensitive.)

Startup and shutdown commands

Startup commands start an OpenEdge session and connect you to a database. The following tables summarize the startup commands for each operating system and their functions. For detailed information on these commands and their parameters, see the descriptions of the commands following the tables. [Table 11–3](#) describes each of the command components.

Table 11–3: OpenEdge command components

(1 of 2)

Component	Description
command	On UNIX, the command runs a script that executes a OpenEdge executable with appropriate parameters.
db-name	Name of the database you want to connect to.

Table 11–3: OpenEdge command components*(2 of 2)*

Component	Description
parameter, qualifier	Operating criteria for the command.
value	Numeric value or file specification for the parameter.

Table 11–4 summarizes the tasks you can perform and the related startup and shutdown commands to use on UNIX systems.

Table 11–4: UNIX startup and shutdown commands*(1 of 2)*

Task	Command
Start a UNIX character Procedure Editor and connect to a single-user database.	<i>pro db-name</i>
Start a UNIX character Procedure Editor and connect to a multi-user database.	<i>mpro db-name</i>
Start a UNIX character client session in batch mode and connect to a single-user database.	<i>bpro db-name</i> <i>-p procedure-name</i>
Start a UNIX OpenEdge character client session in batch mode and connect to a multi-user database.	<i>mbpro db-name</i> <i>-p procedure-name</i>
Start an OpenEdge server-group.	<i>proserve</i> <i>-servergroup server-group-name</i>
Start a server or broker for a multi-user OpenEdge database.	<i>proserve db-name</i> <i>-S service-name</i> <i>-H host-name</i> <i>-N network-type</i>
Shut down a multi-user server or broker for an OpenEdge database.	<i>proshut db-name</i>
Start a remote OpenEdge DataServer broker.	<i>probrkr -S service-name</i> <i>-H host-name</i> <i>-N network-type</i>
Start an asynchronous page writer (APW) for a database. ¹	<i>proapw db-name</i>

Table 11–4: UNIX startup and shutdown commands

(2 of 2)

Task	Command
Start a before-image writer (BIW). ¹	<code>probiw db-name</code>
Start an after-image writer (AIW). ¹	<code>proaiw db-name</code>
Start the OpenEdge Watchdog utility. ¹	<code>prowdog db-name</code>
Shut down a remote OpenEdge DataServer.	<code>proshut db-name</code> <code>-S service-name</code> <code>-H host-name</code> <code>-N network-type</code>
Shut down an APW, AIW, BIW, or Watchdog process. ¹	<code>proshut db-name</code> Choose option 1 (Disconnect a User) to disconnect the process.

1. Option available only on Enterprise product.

Starting single-user OpenEdge in interactive mode

To start single-user OpenEdge, enter the following command:

```
pro [ db-name ] [ parameters ]
```

db-name

Specifies the database you want to start (`-db` is implicit but can be specified).

parameters

Specifies the startup parameters you want to use to describe system and application characteristics. For a detailed description of the OpenEdge startup parameters, see [OpenEdge Deployment: Startup Command and Parameter Reference](#) and [OpenEdge Data Management: Database Administration](#).

Starting single-user OpenEdge in batch or background mode

Batch or background processing is convenient for large-scale database updates or procedures that you can run unattended (for example, at night).

To start single-user OpenEdge in batch or background mode, enter the following command:

```
bpro [ db-name ] -p procedure-name  
[ parameters ] > output-file
```

db-name

Specifies the database you want to start.

-p procedure-name

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

output-file

Specifies the name of the file that receives all output to the default stream.

Redirecting Output

On UNIX you can redirect batch job input and output with the greater than (>) and less than (<) redirection symbols. You can also use the pipe symbol (|) to put a OpenEdge batch run in a command pipeline. See the Batch (-b) startup parameter in [OpenEdge Deployment: Startup Command and Parameter Reference](#) for more information.

In the following example, OpenEdge starts in batch or background mode against the sports database and automatically runs the sportsbat startup procedure. In addition, the system directs output (not otherwise directed) with an OUTPUT TO statement to the file named errlist. For example:

```
bpro sports -p sportsbat.p > errlist
```

Starting the multi-user server or broker

Before you can run multi-user OpenEdge, you must start the multi-user server process. The server process coordinates all the database requests from all the users using a single database. Enter the following command to start the multi-user server process:

```
proserve db-name [ parameters ]
```

db-name

Specifies the database you want to start OpenEdge against (*-db* is implicit).

parameters

Specifies the startup parameters for the broker/server.

The main database server is called the broker. The *broker* process manages shared resources and starts servers for remote users, if necessary. For more information, see the [“Shared-memory configurations”](#) section on page F-2.

Running OpenEdge clients and servers on a network

After setting up your database on the network, you are ready to run OpenEdge. In general, the procedure for running clients and servers on a network of systems is very similar to the procedure for running them on a single system. First, you must start all required database servers and application servers, then start the client sessions that connect to them.

Using network startup parameters

To connect network clients, servers, and application servers, you might have to use a variety of startup parameters to establish and manage network communications among them. The requirements and use of these parameters vary on different operating systems and network environments. For more information of using startup parameters, see [OpenEdge Data Management: Database Administration](#).

Client network parameters

Table 11–5 lists the parameters you use to supply OpenEdge clients with necessary network information.

Table 11–5: Client network parameters

Parameter	Syntax
Host name. ¹	-H
Message buffer size.	-Mm
Network type.	-N
Network version.	-Nv
Service name.	-S

1. For the TCP network type, this required parameter specifies the machine name (address) where the server runs.

Server network parameters

Table 11–6 lists the parameters you use to supply OpenEdge brokers and servers with necessary network information. In an OpenEdge AppServer configuration, use the same parameters to pass information to AppServer brokers and application servers.

Table 11–6: Server network parameters (1 of 2)

Parameter	Syntax
Host name.	-H
Manual server.	-m2
Secondary login broker.	-m3
Maximum clients per server.	-Ma
Minimum clients per server.	-Mi
Maximum dynamic server.	-maxport
Minimum dynamic server.	-minport

Table 11–6: Server network parameters

(2 of 2)

Parameter	Syntax
Message buffer size.	-Mm
AppServer maximum maintained prestart counter.	-Mms
Maximum servers.	-Mn
Maximum servers per protocol.	-Mp
Maximum servers per broker.	-Mpb
AppServer maximum prestart counter.	-Ms
Network type.	-N
Service name.	-S

For more information on the syntax and values for each parameter, see the OpenEdge DataServer Guides (*OpenEdge Data Management: DataServer for Microsoft SQL Server*, *OpenEdge Data Management: DataServer for ODBC*, and *OpenEdge Data Management: DataServer for ORACLE*) and *OpenEdge Deployment: Managing ABL Applications*.

Specifying the network type (-N)

Each OpenEdge executable has a default network type determined by the operating system on which it runs. [Table 11–7](#) lists the default network type that Progress uses on each supported operating system.

Table 11–7: Default network types

Operating system (executable)	Default network type
Windows (client or server)	TCP
UNIX (client or server)	TCP

Network addressing (-S and -H)

In all network environments, you use the Service Name (-S) startup parameter to assign a name to an OpenEdge broker/server. You then address this broker/server from a remote client by using the same value for -S as a startup or database connection parameter. Depending on your network type, you might also have to specify additional addressing criteria for remote OpenEdge clients. In terms of OpenEdge addressing, the TCP protocol uses host addressing.

The TCP protocol requires a remote client to explicitly address the database server machine (or host) on which the server runs. In a TCP network, you must use the Host Name (-H) startup parameter to specify the host address. The -H value is the name assigned to the database server machine in your TCP/IP hosts file.

Starting applications on a network



To start an OpenEdge application on a network:

1. Start each broker or server on its database server machine or application server machine.
2. Start the client applications on the application workstations.

Starting network brokers and servers

You can start most network brokers and servers using either the Progress Explorer (available for Windows systems only) or the PROSERVE command for your database server machine. To use the Progress Explorer, double-click the Progress Explorer icon and follow the directions in the online help for starting brokers and servers. See [Chapter 6, “Administration Utilities,”](#) for more information about starting the Progress Explorer.

Alternatively, in Windows and on UNIX systems you can enter the following command to start brokers for two databases (sports and news) using the TCP network type:

```
proserve sports -S spsrv -H localhost -N TCP -db news -S nwsrv  
-H localhost -N TCP
```

Starting TCP/IP clients in Windows

You can start most network clients using the MPRO command for your application workstation. You can do this either by modifying the Client properties or by entering a command at the command line.

To modify the Client icon, display the properties and modify the shortcut target. Modify the shortcut target with the following parameters to start a client application named `spapp.p` for two databases (sports and news) managed on a host named `dbmach` using the TCP network type:

```
prowin32.exe sports -p spapp.p -S sportssv -H dbmach -N TCP
      -db news -S newssv -H dbmach -N TCP
```

To use the command line in Windows systems, enter the following command to start a client application named `spapp.p` for two databases (sports and news) managed on a host named `dbmach` using the TCP network type:

```
prowin32 sports -p spapp.p -S sportssv -H dbmach -N TCP
      -db news -S newssv -H dbmach -N TCP
```

Starting TCP/IP clients on UNIX

You can start most network clients using the MPRO command for your application workstation. For example, on UNIX machines you can enter the following command to start a client application named `spapp.p` for two databases (sports and news) managed on a host named `dbmach`:

```
mpro sports -p spapp.p -S sportssv -H dbmach -db news -S newssv -H dbmach
```

Starting multiple brokers using the same protocol

You can use either the Progress Explorer or the command-line interface to start multiple brokers that use the same protocol. The `-Mn` parameter and a new parameter, Maximum Servers per Broker (`-Mpb`), determine the number of servers a broker can start. You can use the Progress Explorer to manage and configure server groups.

Using the Progress Explorer to start multiple brokers

You can use the Progress Explorer to start multiple brokers that use the same protocol. Start the Progress Explorer by double-clicking the Progress Explorer Tool icon in the OpenEdge program group. Follow the instructions in the Progress Explorer online help to start and configure brokers. See [Chapter 10, “Configuration,”](#) for more information about starting the Progress Explorer.

Using the command-line interface to start multiple brokers

Use the following commands to start two brokers that use TCP and start multiple servers each:

```
proserve db-name -S server-name -N network-type -H host-name -Mnn -Mpb n  
proserve db-name -S server-name -N network-type -H host-name -Mpbm -m3
```

`db-name`

Specifies the database you want to start. If the database is not in the current directory, you must specify the full pathname of the database.

`-S service-name`

Specifies the database server or broker process service name. You must specify the service name in a TCP network.

`-N network-type`

Specifies the network protocol, which is TCP.

`-H host-name`

Specifies the machine where the database server runs.

`-Mn n`

Specifies the maximum number of remote client servers and login brokers that the broker process can start.

`-Mpb n`

Specifies the number of servers that the login broker can start to serve remote users. This applies to the login broker that is being started.

`-m3`

Starts the secondary login broker.

You would enter the following commands to start two brokers that use TCP and start four servers each:

```
proserve db -S demosv1 -N tcp -H myhost -Mn 9 -Mpb 4
proserve db -S demosv2 -N tcp -H myhost -Mpb 4 -m3
```

As the example shows, the `-Mn` value must be large enough to account for each additional broker and all servers. If you do not specify `-Mpb`, the value of `-Mn` becomes the default.

You must include the `-m3` parameter with every secondary broker startup command. While the `-Mpb` sets the number of servers a broker can start, the `-m3` parameter actually starts the secondary broker.

If you start multiple brokers, you should also run the Progress Watchdog process (PROWDOG). PROWDOG enables you to restart a dead secondary broker without shutting down the database server.

Accessing a server behind a firewall

OpenEdge allows you to use the Minimum Dynamic Server Port (`-minport`) and the Maximum Dynamic Server Port (`-maxport`) server startup parameters to provide client access to an OpenEdge server that is behind a firewall. This communication is possible only when the access to the server can be limited. You supply this limit when you specify a group of port numbers with the `-minport` and `-maxport` parameters.

For example, suppose you start the following two login brokers:

```
proserve db -S demosv1 -N tcp -H myhost -minport 4000 -maxport 4040
proserve db -S demosv2 -N tcp -H myhost -minport 4041 -maxport 4080 -m3
```

A client requesting a connection from the first broker, `demosv1`, is assigned a port number in the range of 4000 to 4040. The 4000-to-4040 range limits access to the server by limiting communication to just 40 ports.

The default for `-minport` is 1025 for all platforms. Ports lower than 1025 are usually reserved for system TCP and UDP. The default for `-maxport` is 2000 for all platforms. Remember that some operating systems choose transient client ports in the 32768-to-65535 range. Choosing a port in this range might produce unwanted results.

Starting and running multi-user OpenEdge in interactive mode in Windows

Enter the following command to start and run Progress in interactive mode:

```
prowin32 [ db-name ] [ parameters ]
```

db-name

Specifies the database you want to start. If the database is not in the current directory, you must specify the full pathname of the database.

parameters

Specifies the startup parameters you want to use.

The database you name when starting multi-user Progress must be in the current directory, or you must specify the full pathname of the database. For example, if you are using UNIX and you log in as `sue`, the login directory is `/usr/sue`.

Starting and running multi-user OpenEdge in interactive mode on UNIX

Enter the following command to start and run OpenEdge in interactive mode:

```
mpro [ db-name ] [ parameters ]
```

db-name

Specifies the database you want to start. If the database is not in the current directory, you must specify the full pathname of the database.

parameters

Specifies the startup parameters you want to use.

On UNIX, the MPRO command starts either a local or remote client. If the Host Name (-H) and Service Name (-S) parameters are supplied, OpenEdge starts a remote client—a client that is assigned to a server. Otherwise, OpenEdge starts a local self-service client. Note that specifying -H and -S when starting a client on the local host machine actually produces a “local remote client” (a local process that accesses the database through a server).

The database you name when starting multi-user OpenEdge must be in the current directory, or you must specify the full pathname of the database. For example, if you are using UNIX and you log in as sue, the login directory is `/usr/sue`.

Starting and running multi-user OpenEdge clients in batch or background mode in Windows

Before you can start a multi-user OpenEdge batch or background job, you must start the server for the database you want to use. You can start the server either by modifying the Client icon properties or by typing a command at the command line.

Using the Client to start multi-user OpenEdge in batch or background mode

You can modify the Client properties to start multi-user OpenEdge in batch or background mode. To modify the Client icon, display the properties and modify the shortcut target with the following parameters:

```
prowin32 [ database name ] -N network-type  
-S service-name -p procedure-name  
[ parameters ]
```

Using the command-line interface to start multi-user OpenEdge in batch or background mode

To use the command-line interface to start multi-user OpenEdge in batch or background mode, enter the following command:

```
prowin32 [ database name ] -N network-type  
-S service-name -p procedure-name  
[ parameters ]
```

database-name

Specifies the database you want to start.

-p procedure-name

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

error-file

Specifies the file where error messages are sent.

Using OpenEdge Explorer to start multi-user OpenEdge in batch or background mode

You can use the Progress Explorer to create a task that starts multi-user OpenEdge in batch or background mode in a Windows system. See [Chapter 10, “Configuration,”](#) for information on starting the Progress Explorer. See the Progress Explorer online help for instructions on starting Windows multi-user OpenEdge in batch or background mode.

Starting and running multi-user OpenEdge clients in batch or background mode on UNIX

Before you can start a multi-user OpenEdge batch or background job, you must start the server for the database you want to use.

To start multi-user OpenEdge in batch or background mode, enter the following command:

```
mbpro db-name -p procedure-name [ parameters ]  
> error-file
```

db-name

Specifies the database you want to start.

-p procedure-name

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

error-file

Specifies the file where error messages are sent.

Part III

OpenEdge Products and Components

[Chapter 12, OpenEdge Installation Products and Components in Windows](#)

[Chapter 13, OpenEdge Installation Products and Components on UNIX](#)

[Appendix A, Preinstallation Checklist for Windows](#)

[Appendix B, Preinstallation Checklist for UNIX](#)

[Appendix C, Command and Utility Reference](#)

[Appendix D, OpenEdge National Language Support](#)

[Appendix E, NameServer and Name Server Load Balancing Details](#)

[Appendix F, Configuration Models](#)

[Appendix G, AdminServer Authorization and Authentication](#)

OpenEdge Installation Products and Components in Windows

When you install OpenEdge you can choose from two installation options: complete or custom. This chapter provides you with a list of the components and subcomponents that you install for each product when you choose a complete installation, as described in the following sections:

- [OpenEdge Release product headlights](#)
- [OpenEdge installation options](#)
- [OpenEdge product components and subcomponents](#)

OpenEdge Release product highlights

The following sections briefly identifies some specific OpenEdge products.

OpenEdge Architect

The OpenEdge Architect is a development environment based on the Eclipse platform. For information about the OpenEdge Architect, see the OpenEdge Architect Guide, an online help system, packaged with and available from the OpenEdge Architect product. For details about OpenEdge Architect components and subcomponents, see [Table 12–9](#).

Audit Policy Maintenance (APM)

Audit Policy Maintenance is a core business services auditing framework. Among the many functional elements of this framework is the Audit Policy Maintenance (APM) which enables you to create and maintain auditing policies for applications and databases. Application and database auditing include activities such as data collection, configuration, management, and security.

The Audit Policy Maintenance (APM) is available as a subcomponent of the OpenEdge Personal RDBMS, OpenEdge Workgroup RDBMS, OpenEdge Enterprise RDBMS and other OpenEdge products.

For more information about the auditing framework and the APM, see [OpenEdge Getting Started: Core Business Services](#).

OpenEdge Adapter for Sonic ESB

The OpenEdge Adapter for Sonic ESB is used to integrate an OpenEdge application server and the Sonic Enterprise Service Bus. The adapter consists of Java JAR files and configuration files. It uses a standard OpenEdge installation procedure but does not require a complete OpenEdge environment.

The OpenEdge installation program installs a Sonic ESB container in which the OpenEdge Adapter for Sonic ESB will run. The Domain Manager for a Sonic installation requires information from this container, and the OpenEdge installation program uses the information you provide to connect to the Domain Manager and obtain this information.

Note: If the Domain Manager container is not running when you are performing the OpenEdge installation, or the Installation Utility cannot connect to the Domain Manager, a script file, either `offline.bat` or `offline.sh`, is created to hold the required information. You can run either file once the Domain Manager is running.

For more information about the OpenEdge Adapter for Sonic ESB, see *OpenEdge Development: Messaging and ESB*.

Web Services Adapter

A *Web service* is an application that can be accessed over the Internet (or an intranet) using industry-standard protocols. Web services provide an industry-standard way for all types of client applications to call functions on all types of application servers, over any network configuration that supports SOAP, and where the application program interface (API) can be described using Web Services Descriptor Language (WSDL).

In OpenEdge, you can build and deploy Web services using the AppServer, Web service tools, and the Web Services Adapter (WSA). You can define Web services using the Open Client Toolkit much as you define traditional Open Client proxies. However, instead of generating Open Client proxies for immediate access by Open Client applications, you generate a client interface definition, and deploy this definition as a Web service that you then manage within the context of the OpenEdge Web Services Adapter.

The WSA is a Java servlet running on a Web server or stand-alone JSE. This Java servlet serves as the SOAP-to-business interface gateway and Web service management engine for your Web service. Thus, the WSA provides users access to the Web service WSDL file and supports Web service administration; it also manages the exchange of Web service SOAP requests and generates SOAP responses between Web service clients and AppServers at run time.

You can install the WSA as a separate stand-alone product on its own machine. It is available as a separately installable OpenEdge component, as a sub-component of the OpenEdge Application Server Enterprise product and of the OpenEdge Development Server product, and as a download from the Progress Software Corporation Web site at <http://www.progress.com/esd>. For information on WSA configuration, see the WSA sections of *OpenEdge Development: Web Services* and *OpenEdge Application Server: Administration*.

OpenEdge installation options

You can choose between two options when installing OpenEdge: complete or custom. These installation options allow you to choose the option that is best for you, depending on how many products you are installing, which product components are mandatory, and which are optional, and whether all the products reside on the same system.

Complete installation option

When you choose the complete installation option and specify the products you want to install, all mandatory, recommended, and optional components and subcomponents are installed automatically. For this reason, a complete installation usually requires more disk space than a custom installation requires.

Custom installation option

When you choose the custom installation option, all mandatory products and subcomponents are installed, but you can selectively install the recommended and optional components and subcomponents on a product-by-product basis. A custom installation provides more advanced users, for whom this method is recommended, a means to distribute OpenEdge components on different machines, select product components to suit their business needs, and work around issues such as disk space limitations.

Caution: Removing recommended product components and/or subcomponents can affect the functionality of a product.

The mandatory, recommended, and optional components and subcomponents for each OpenEdge product are listed, by product, in the [“OpenEdge product components and subcomponents”](#) section on page 12–5.

For a description of the steps to follow when installing OpenEdge, see the OpenEdge online installation help system. Online help is accessible from all supported platforms in Windows and on UNIX.

OpenEdge product components and subcomponents

The tables in the following sections list the components and subcomponents that are installed for each product.

4GL Development System

[Table 12–1](#) lists the 4GL Development System components and subcomponents. When you choose the complete installation option and install 4GL Development, all components and subcomponents listed are installed.

Table 12–1: 4GL Development System components and subcomponents

(1 of 5)

Component	M/R/O ¹	Subcomponent	M/R/O
4GL Client	M	Base Client — 4GL	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
ADE Source Code	M	ADE Common Source	M
		ADM Source	M
		DB Administration Source	M
		Editor Source	M
		ProTools Source Code	M
Application Debugger	R	Application Debugger	R
		Remote Debugging	M

Table 12–1: 4GL Development System components and subcomponents*(2 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Character Base Tools	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—
Character Runtime Client — Dev	M	—	—
Client-Side Web Service	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
4GL utilities	R	XSD-4GL	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	M	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M
DataDirect ODBC Driver Support	R	—	—

Table 12–1: 4GL Development System components and subcomponents*(3 of 5)*

Component	M/R/O¹	Subcomponent	M/R/O
Graphical Base Tools	M	ADM Runtime — GUI	M
		Advanced Editing	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
Name Server	M	—	—
NetSetup	O	—	—
OE Build Utility	R	—	—
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for Sonic MQ	R
		OpenEdge Adapter for Sonic ESB	R
		Proxy Generator	M
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Service Schema	R
Oracle DataServer Client	O	—	—

Table 12–1: 4GL Development System components and subcomponents*(4 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	Language subset	O
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL JBDC Client	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL OBDC Client	M
		SQL Server	M
Report Engine	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M

Table 12–1: 4GL Development System components and subcomponents*(5 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
SQL Database Server	O	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		JDK	M
		Progress Databases	M
		SQL Server	M
Toolkit	M	—	—

1. M=Mandatory, R=Recommended, O=Optional

AppServer Internet Adapter (AIA)

Table 12–2 lists the AppServer Internet Adapter components that are installed when you choose the Complete installation option.

Table 12–2: AppServer Internet Adapter (AIA) components and subcomponents*(1 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Internet Adapter	M	—	—
Common Broker	M	—	—
Common Files (minimum)	M	—	—
Java Server	M	Java Server	M
Name Server	M	—	—
Progress Messages (PROMSGS)	M	Language subset	O

Table 12–2: AppServer Internet Adapter (AIA) components and subcomponents *(2 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

1. M=Mandatory, R=Recommended, O=Optional

Client Networking

[Table 12–3](#) lists the Client Networking components and subcomponents that are installed when you choose the Complete installation option and install Client Networking.

Table 12–3: Client Networking components and subcomponents *(1 of 4)*

Component	M/R/O ¹	Subcomponent	MR/O
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Character Base Tools	O	ADM Runtime — CHAR	M
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Base Tools — Optional	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—
Character Runtime Client — Dev	M	—	—

Table 12–3: Client Networking components and subcomponents*(2 of 4)*

Component	M/R/O¹	Subcomponent	MR/O
Client Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
DataDirect ODBC Driver Support	R	–	–
Database Administration Tools	M	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administrations	M
Graphical Base Tools — Client	M	ADM Runtime — GUI	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
Name Server	M	–	–
NetSetup	O	–	–
OE Build Utility	R	–	–

Table 12–3: Client Networking components and subcomponents*(3 of 4)*

Component	M/R/O ¹	Subcomponent	MR/O
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Ext	M
		Java Client Support	R
		Java Server	M
		Java Class Tailoring	M
		OE Adapter for Sonic MQ	R
		Web Services Schema	R
Oracle DataServer Client	O	—	—
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL JDBC Client	M
		SQL Common	M
		SQL Server	M
		SQL JDBC Client	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL ODBC Client	M

Table 12–3: Client Networking components and subcomponents*(4 of 4)*

Component	M/R/O ¹	Subcomponent	MR/O
Progress Messages (PROMSGS)	M	Language subset	O
Report Engine	M	—	—
Remote Debugging	M	—	—
Runtime Client	M	Base Client—RT	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M

1. M=Mandatory, R=Recommended, O=Optional

NameServer

Table 12–4 lists the NameServer components and subcomponents.

Table 12–4: NameServer components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
NameServer	M	–	–
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	Language subset	O

1. M=Mandatory, R=Recommended, O=Optional

NameServer Load Balancer

Table 12–5 lists the NameServer Load Balancer components that are installed when you choose the Complete installation option and install the NameServer Load Balancer option.

Table 12–5: NameServer Load Balancer components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common files	M	Common Files	M
		WebSpeed Common	M

1. M=Mandatory, R=Recommended, 0=Optional

OpenEdge Adapter for Sonic ESB

Table 12–6 lists the Sonic ESB Adapter components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed.

Table 12–6: OpenEdge Adapter for Sonic ESB components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Java Class Tailoring	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
OpenEdge Adapter for Sonic ESD	M	–	–

1. M=Mandatory, R=Recommended, 0=Optional

OpenEdge Application Server — Basic

Table 12–7 lists the Application Server Basic components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed.

Table 12–7: OpenEdge Application Server — Basic components and subcomponents (1 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Runtime Client	M	–	–
Basic Server Option	R	ADE Common Source	M
		ADM Runtime — GUI	R
		ADM Runtime — Char	M
		AppServer — Basic	R
		Auditing Policy Maintenance	M
		Base Client — 4GL	M
		Base ADE	M
		Character Client — WebSpeed	R
		Common Broker	M
		Crypto Tools	M
		Editor Source	M
		Graphical Client	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — Char	M
		Progress Databases	M
		SQL Server	M
		Transaction Server — Basic	R
		Web Static	M
		WebSpeed Messenger	R
		WebSpeed Run-time	M
		WebSpeed Tools	M

Table 12–7: OpenEdge Application Server — Basic components and subcomponents (2 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Basic Server Option (cont.)	R	XML	M
Client-Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
OE Build Utilities	R	—	—
OE Perl	M	—	—
Progress Messages (PROMSGS)	M	Language	O
Remote Debugging	M	—	—
Open Client Adapter Options — Basic	R	OpenEdge Adapter for Sonic MQ	R
		AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Class Tailoring	M
		Java Server	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

Table 12–7: OpenEdge Application Server — Basic components and subcomponents (3 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Server Data Source Options	O	DataDirect ODBC Driver Support	O
		Database Tools	M
		ICU PSC	M
		Oracle Client	O
		ESQL Client	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M
Server Admin and Configuration	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M
		WebSpeed Tools	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Application Server — Enterprise

Table 12–8 lists the Application Server Basic components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed.

Table 12–8: OpenEdge Application Server — Enterprise components and subcomponents

(1 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Runtime Client	M	—	—
Client-Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Enterprise Server Options	R	AppServer — Enterprise	R
		ADE Common Source	M
		ADM Runtime — Char	M
		ADM Runtime — GUI	R
		Base Client — 4GL	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Character Client — WebSpeed	R
		Client-Side Security	R

Table 12–8: OpenEdge Application Server — Enterprise components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Enterprise Server Options (cont.)	R	Common Broker	M
		Crypto Tools	M
		Editor Source	M
		Graphical Client	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — Char	M
		Progress Databases	M
		Security Common	M
		SQL Server	M
		Transaction Server — Enterprise	R
		Web Static	M
		WebSpeed Messenger	R
		WebSpeed Run-time	M
		WebSpeed Tools	M
		XML	M
OE Build Utility	R	—	—
OE Perl	M	—	—
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—

Table 12–8: OpenEdge Application Server — Enterprise components and subcomponents*(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Open Client Adapter Options — Enterprise	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for Sonic MQ	R
		OpenEdge Adapter for Sonic ESB	R
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Services Schema	R
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
Server Data Source Options	O	DataDirect ODBC Driver Support	O
		Database Tools	M
		Oracle Client	O
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

Table 12–8: **OpenEdge Application Server — Enterprise components
and subcomponents**

(4 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
Server Admin and Configuration	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M
		WebSpeed Tools	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Architect

Table 12–9 lists the OpenEdge Architect components and subcomponents. When you choose the Complete installation option and install OpenEdge Architect, all components and subcomponents listed are installed.

Table 12–9: OpenEdge Architect components and subcomponents

(1 of 6)

Component	M/R/O ¹	Subcomponent	M/R/O
Application Server options	R	4GL Database	M
		4GL Server	M
		ADM Runtime — GUI	R
		ADM Runtime — CHAR	M
		AppServer — Dev	R
		Base Client — 4GL	M
		Character Client — WebSpeed	R
		Common Broker	M
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — CHAR	R
		Progress Databases	M
		SQL Server	M
		Transaction Server — Dev	R
		WebSpeed Messenger	R
		Web Static	M
		WebSpeed Run-time	M
		WebSpeed Tools	M
		XML	M

Table 12–9: OpenEdge Architect components and subcomponents*(2 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Client-Side Web Services	R	Client-Side Security	R
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
4GL utilities	R	XSD-4GL	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Development Data Source Option	O	4GL Server	M
		DataDirect ODBC Driver Support	O
		Database Utilities	M
		Database Tools	M
		Database Server	M
		ESQL Client	M
		ICU PSC	M
		Oracle Client	O
		JDK	M
		SQL Server	M
		Progress Databases	M
		SQL ODBC Client	M
		SQL JDBC Client	M
		SQL Common	M

Table 12–9: OpenEdge Architect components and subcomponents*(3 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge Architect Development	R	4GL Server	M
		ActiveX Control Development	M
		ActiveX Control Runtime	M
		ADM GUI Runtime	R
		ADM Runtime CHAR	M
		Auditing Policy Maintenance	M
		Base Client — 4GL	M
		Character Client — RT	O
		Character Image	O
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		Graphical Client	M
		ICU PSC	M
		OpenEdge Architect	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		JDK	M
		Procedure Editor — GUI	M
		Progress Databases	M
		Proxy Generator	M
		Remote Debugging	M
		Web Static	M
		WebClient Assembler Utility	R
		WebClient Client	M
		WebSpeed Runtime	M

Table 12–9: OpenEdge Architect components and subcomponents*(4 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge Architect Development (cont.)	R	XML	M
		Application Debugger	R
		Progress Dynamics — OE Architect	O
		Progress Dynamics RT — OE Architect	O
OpenEdge Architect AppBuilder	R	Advanced Editing	M
		APPBuilder Core	M
		Base ADE	M
		Compile Tool — CHAR	O
		WebSpeed Workshop-Dev	R
		Compile Tool — GUI	R
NetSetup	O	—	—
OEBuild Utility	R	—	—
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		DotNET Client support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for Sonic ESB	R
		OpenEdge Adapter for Sonic MQ	R
		Proxy Generator	M
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Services Schema	R

Table 12–9: OpenEdge Architect components and subcomponents*(5 of 6)*

Component	M/R/O¹	Subcomponent	M/R/O
Other Options	O	Base ADE	M
		Client-Side Security	R
		Report Builder Engine	M
		Results (Graphical)	O
		Security Common	M
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

Table 12–9: OpenEdge Architect components and subcomponents*(6 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Studio Admin and Configuration	R	4GL Database	M
		Administration Server	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Character Administration	R
		Database Utilities	M
		NameServer	R
		Common Broker	M
		UBroker Tools	M
		WebSpeed Tools	M
		NameServer	R
		ToolKit	R
		Administration Server	M
		Explorer Tools	M
		Graphical Administration	M
		Java Ext	M
		Java Server	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge DataServer for MS SQL Server

Table 12–10 lists the OpenEdge DataServer for MS SQL Server components and subcomponents. The DataServer for Microsoft SQL Server is compatible with Microsoft SQL Server 2000 and later. Choosing the Complete option results in the installation of all components and subcomponents listed.

Table 12–10: OpenEdge DataServer — MS SQL Server components and subcomponents

(1 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
ActiveX Control support	M	ActiveX Control Runtime Support	M
Character Base Tools	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—
Character Runtime Client — Dev	O	—	—
Common files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	M	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration	M
Graphical Base Tools	M	ADM Runtime — GUI	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Procedure Editor — GUI	M

Table 12–10: OpenEdge DataServer — MS SQL Server components and subcomponents*(2 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
MS SQL Server DataServer	M	Broker	M
		MS SQL Server DataServer	M
MS SQL Server DataServer Drivers	M	—	—
Name Server	M	—	—
NetSetup	O	—	—
OE Build Utility	R	—	—
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—

Table 12–10: OpenEdge DataServer — MS SQL Server components and subcomponents*(3 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Runtime Client	M	Base Client — RT	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
Schema Holder and Server	M	4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge DataServer for ODBC

Table 12–11 lists the OpenEdge DataServer for ODBC components and subcomponents that are installed when you choose the Complete installation option and install the OpenEdge DataServer for ODBC.

Table 12–11: OpenEdge DataServer — ODBC components and subcomponents*(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Character Base Tools	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—

Table 12–11: OpenEdge DataServer — ODBC components and subcomponents (2 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Character Runtime Client — Dev	O	—	—
Common files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	M	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration	M
Graphical BaseTools	M	ADM Runtime — GUI	M
		Base ADE	M
		Compile Tools — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
Name Server	M	—	—
NetSetup	O	—	—
ODBC DataServer	M	Broker	M
		ODBC DataServer	M
ODBC DataServer Drivers	M	—	—
OE Build Utility	R	—	—
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R

Table 12–11: OpenEdge DataServer — ODBC components and subcomponents *(3 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Remote Debugging	M	–	–
Runtime Client	M	Base Client — RT	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
Secure Clients	M	ClientSide Security	R
		Perl	M
		Security Common	M
Schema Holder and Server	M	4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge DataServer for Oracle

Table 12–12 lists the OpenEdge DataServer for Oracle components and subcomponents that are installed when you choose the Complete installation option and install the Oracle DataServer.

Table 12–12: OpenEdge DataServer for Oracle components and subcomponents *(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Character BaseTools	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—
Character Runtime Client — Dev	O	—	—
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	M	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Graphical Base Tools	M	ADM Runtime — GUI	M
		Base ADM	M
		Compile Tools — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
Name Server	M	—	—
NetSetup	O	—	—
OE Build Utility	R	—	—

Table 12–12: OpenEdge DataServer for Oracle components and subcomponents (2 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Class Tailoring	M
		Java Server	M
		OpenEdge Adapter for SonicMQ	R
Oracle DataServer	M	Broker	M
		Oracle DataServer	M
Oracle DataServer Client	O	—	—
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Runtime Client	M	Base Client — RT	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M

Table 12–12: OpenEdge DataServer for Oracle components and subcomponents (3 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Schema Holder and Server	M	4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Development Server

Table 12–13 lists the OpenEdge Development Server components and subcomponents. When you choose the Complete installation option and install the OpenEdge Development Server, all components and subcomponents listed are installed.

Table 12–13: OpenEdge Development Server components and subcomponents (1 of 5)

Component	M/R/O ¹	Subcomponent	M/R/O
Administration and Configuration	M	Administration Server	M
		Auditing Policy Maintenance	M
		Graphical Administration	M
		Base ADE	M
		Character Administration	M
		Common Broker	M
		Database Utilities	M
		4GL Database	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M

Table 12–13: OpenEdge Development Server components and subcomponents*(2 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Administration and Configuration (cont.)	M	WebSpeed Tools	M
Client Side Web Services	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
4GL utilities	R	XSD — 4GL	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Development Data Source Options	O	4GL Server	M
		Data Direct ODBC Driver Support	O
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ESQL Client	M
		ICU PSC	M
		JDK	M
		Oracle Client	O
		Progress Databases	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M
NetSetup	O	—	—
OE Build Utility	R	—	—
OE Perl	M	—	—

Table 12–13: OpenEdge Development Server components and subcomponents *(3 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Development Server Options	R	ADM Run-time GUI	R
		ADM Run-time CHAR	M
		AppBuilder Core	M
		AppServer — Dev	R
		4GL Database	M
		4GL Server	M
		Graphical Client	M
		ICU PSC	M
		Name Server	R
		Procedure Editor — Char	M
		Progress Databases	M
		Security Common	M
		Base Client — 4GL	M
		Base ADE	M
		Character Client — WebSpeed	R
		Client-Side Security	R
		Common Broker	M
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		DB Administration Source	M

Table 12–13: OpenEdge Development Server components and subcomponents (4 of 5)

Component	M/R/O ¹	Subcomponent	M/R/O
Development Server Options (cont.)	R	Desktop	M
		Editor Source	M
		SQL Server	M
		WebSpeed Messenger	R
		Web Static	M
		WebSpeed Run-time	M
		WebSpeed Tools	M
		XML	M
		Transaction Server-Dev	R
		Progress Dynamics RT	R
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		DotNET Client Support	R
		Proxy Generator	M
		OpenEdge Adapter for SonicMQ	R
		OpenEdge Adapter for Sonic ESB	R
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Services Schema	R
Progress Messages (PROMSGS)	M	Language Subset	O
Server Source Code Options	R	ADE Common Source	O
		ProTools Source Code	M
		Editor Source	O

Table 12–13: OpenEdge Development Server components and subcomponents (5 of 5)

Component	M/R/O ¹	Subcomponent	M/R/O
Toolkit	M	—	—
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
Other Development Server Options	R	ADM Runtime — CHAR	O
		Application Debugger	R
		Character Image	O
		Character Client — Runtime	O
		Compile Tool — CHAR	O
		Crypto Tools	M
		Procedure Editor — CHAR	O
		Remote Debugging	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Enterprise RDBMS

Table 12–14 lists the Enterprise Database components and subcomponents that are installed when you choose the Complete installation option and install the Enterprise Database.

Table 12–14: OpenEdge Enterprise RDBMS components and subcomponents

(1 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
4GL	O	4GL Server	M
		ActiveX Control Runtime Support	M
		ADM Runtime — CHAR	O
		ADM Runtime — GUI	O
		Auditing Policy Maintenance	M
		Base ADE	M
		Base Client — DA	M
		Character Admin	O
		Character Client — 4GL	O
		Character Image	O
		Compile Tool — GUI	O
		Compile Tool — CHAR	O
		Crypto Tools	M
		Desktop	M
		Graphical Administration	M
		Graphical Client	M
		Oracle Client	O
		Procedure Editor — CHAR	O
		Procedure Editor — GUI	M
		Report Engine	M
		SQL Server	M
		ICU PSC	M
		XML	M

Table 12–14: OpenEdge Enterprise RDBMS components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Client Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Failover Clusters	R	Cluster Common	M
NameServer	M	–	–
NetSetup	O	–	–
OE Build Utility	R	–	–
OE Perl	M	–	–
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
DB Management	M	4GL Server	M

Table 12–14: OpenEdge Enterprise RDBMS components and subcomponents*(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
DB Management (cont.)	M	Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		Progress Databases	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
		SQL Server	M
Progress Explorer Tools — DB	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
Progress Messages (PROMSGS)	M	All Language subset	O
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQLClient	M
		ICU PSC	M
		SQLCommon	M
		SQL Server	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL JDBC Client	M
OpenEdge SQL ODBC Clients	O	Database Tools	M

Table 12–14: OpenEdge Enterprise RDBMS components and subcomponents (4 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge SQL ODBC Clients (cont.)	O	ICU PSC	M
		SQL ODBC Client	M
		SQL Server	M
		SQL Common	M
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
SQL	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Personal RDBMS

Table 12–15 lists the Personal Database components and subcomponents that are installed when you choose the Complete installation option and install the Personal RDBMS.

Table 12–15: OpenEdge Personal RDBMS components and subcomponents

(1 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
4GL ²	O	4GL Server	M
		ActiveX Control Runtime Support	M
		ADM Runtime — CHAR	O
		ADM Runtime — GUI	O
		Base ADE	M
		Auditing Policy Maintenance	M
		Base Client — DA	M
		Character Administration	O
		Character Client — 4GL	O
		Character Image	O
		Compile Tool — CHAR	O
		Compile Tool — GUI	O
		Crypto Tools	M
		Desktop	M
		Graphical Administration	M
		Graphical Client	M
		Oracle Client	O
		Procedure Editor — CHAR	O
		Procedure Editor — GUI	M
		Report Engine	M
		ICU PSC	M
		SQL Server	M
		XML	M

Table 12–15: OpenEdge Personal RDBMS components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Client-Side Web Services	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common files	M	Common Files	M
		WebSpeed Common	M
DB Management	M	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		Progress Databases	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
		SQL Server	M
Name Server	M	—	—
NetSetup	O	—	—
OE Build Utility	R	—	—
OE Perl	M	—	—

Table 12–15: OpenEdge Personal RDBMS components and subcomponents*(3 of 4)*

Component	M/R/O¹	Subcomponent	M/R/O
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL JDBC Client	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL ODBC Client	M

Table 12–15: OpenEdge Personal RDBMS components and subcomponents*(4 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Progress Explorer Tools — DB	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
SQL	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

2. The 4GL component of the OpenEdge Personal RDBMS includes the Client Networking functionality.

OpenEdge Replication

Table 12–16 lists the OpenEdge Replication components and subcomponents that are installed when you choose the Complete Installation option.

Table 12–16: OpenEdge Replication components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Progress Messages (PROMSGS)	M	Language subset	O
Replication	M	Replication Common	M
		Replication Installation	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Replication Plus

Table 12–17 lists the OpenEdge Replication Plus components and subcomponents that are installed when you choose the Complete Installation option.

Table 12–17: OpenEdge Replication Plus components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Progress Messages (PROMSGS)	M	Language subset	O
Replication	M	Replication Common	M
		Replication Installation	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge SQL Client Access

Table 12–18 lists the OpenEdge SQL Client Access components and subcomponents.

Table 12–18: OpenEdge SQL Client Access components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Name Server	M	–	–
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL JDBC clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL JDBC Client	M
OpenEdge SQL ODBC clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
		SQL ODBC Client	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Studio

Table 12–19 lists the OpenEdge Studio components and subcomponents.

Table 12–19: OpenEdge Studio components and subcomponents (1 of 6)

Component	M/R/O ¹	Subcomponent	M/R/O
Application Server Options	R	4GL Database	M
		4GL Server	M
		ADM Runtime — GUI	R
		ADM Runtime — CHAR	M
		AppServer — Dev	R
		Base Client — 4GL	M
		Character Client — WebSpeed	R
		Common Broker	M
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		Editor Source	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — CHAR	R
		Progress Databases	M
		SQL Server	M
		Transaction Server — Dev	R
		WebSpeed Messenger	R
		Web Static	M
		WebSpeed Run-time	M
		WebSpeed Tools	M
		XML	M
Client-Side Web Services	R	Client-Side Security	R

Table 12–19: OpenEdge Studio components and subcomponents*(2 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Client-Side Web Services (cont.)	R	Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
4GL utilities	R	XSD — 4GL	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Development Data Source Option	O	4GL Server	M
		DataDirect ODBC Driver Support	O
		Database Utilities	M
		Database Tools	M
		Database Server	M
		ESQL Client	M
		ICU PSC	M
		Oracle Client	O
		JDK	M
		SQL Server	M
		Progress Databases	M
		SQL ODBC Client	M
		SQL JDBC Client	M
		SQL Common	M
Development Source Code Option	R	ADE Common Source	M
		ADM Source	R
		DB Administration Source	O
		Editor Source	O
		ProTools Source Code	M
NetSetup	O	—	—
OE Build Utility	R	—	—

Table 12–19: OpenEdge Studio components and subcomponents*(3 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for Sonic ESB	R
		OpenEdge Adapter for Sonic MQ	R
		Proxy Generator	M
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Services Schema	R
Progress Messages (PROMSGS)	M	Language subset	O

Table 12–19: OpenEdge Studio components and subcomponents*(4 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Studio Admin and Configuration	R	Administration Server	M
		Character Administration	R
		4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Common Broker	M
		Database Utilities	M
		Graphical Administration	M
		Java Ext	M
		Java Server	M
		Administration Server	M
		Explorer Tools	M
		Name Server	R
		Ubroker Tools	M
		Toolkit	R
		WebSpeed Tools	M
Studio Development	R	4GL Server	M
		ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
		ADM Runtime — CHAR	M
		ADM Run-time GUI	R
		Advanced Editing	M
		AppBuilder Core	M
		Application Debugger	R
		Auditing Policy Maintenance	M
		Base Client — 4GL	M
		Character Client — Runtime	O

Table 12–19: OpenEdge Studio components and subcomponents*(5 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Studio Development (cont.)	R	Character Image	O
		Compile Tool — CHAR	O
		Compile Tool — GUI	R
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		DB Admin Source	M
		Desktop	M
		Editor Source	M
		Graphical Client	M
		ICU PSC	M
		Java Ext	M
		Java Server	M
		Base ADE	R
		JDK	M
		Java Client Support	R
		Procedure Editor — CHAR	O
		Procedure Editor — GUI	M
		Progress Dynamics	R
		Progress Dynamics RT	R
		Progress Databases	M
		Proxy Generator	M
		Remote Debugging	M
		Web Static	M
		WebClient Assembler Utility	R
		WebSpeed Runtime	M
		WebClient — Client	M

Table 12–19: OpenEdge Studio components and subcomponents*(6 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Studio Development (cont.)	R	WebSpeed Workshop — Dev	R
		XML	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
Other Options	O	Base ADE	M
		Client-Side Security	R
		Report Builder Engine	M
		Results (Graphical)	O
		Security Common	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Workgroup RDBMS

Table 12–20 lists the Workgroup RDBMS components and subcomponents. When you choose the Complete installation option and install the Workgroup RDBMS, all components and subcomponents listed are installed.

Table 12–20: OpenEdge Workgroup components and subcomponents*(1 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
4GL	O	4GL Server	M
		ActiveX Control Runtime Support	M
		ADM Runtime — CHAR	O
		ADM Runtime — GUI	O
		Auditing Policy Maintenance	M

Table 12–20: OpenEdge Workgroup components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
4GL (cont.)	O	Character Image	O
		Compile Tool — CHAR	O
		Compile Tool — GUI	O
		Crypto Tools	M
		Desktop	M
		Graphical Administration	M
		Graphical Client	M
		Oracle Client	O
		Procedure Editor — CHAR	O
		Procedure Editor — GUI	M
		Report Engine	M
		ICU PSC	M
		Base ADE	M
		Base Client — DA	M
		Character Administration	O
		Character Client — 4GL	O
		SQL Server	M
		XML	M
Client-Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common files	M	Common Files	M
		WebSpeed Common	M
Name Server	M	—	—
NetSetup	O	—	—

Table 12–20: OpenEdge Workgroup components and subcomponents*(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
OE Build Utility	R	–	–
OE Perl	M	–	–
DB Management	M	Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		4GL Server	M
		Progress Databases	M
		SQL Server	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
Progress Explorer Tools — DB	M	Administration Server	M
		Common Broker	M
		Explorer Tools	M
		Java Ext	M
		Java Server	M
Progress Messages (PROMSGS)	M	Language subset	O

Table 12–20: OpenEdge Workgroup components and subcomponents*(4 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL ODBC Client	M
OpenEdge ESQL/C Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		ESQL Client	M
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
SQL	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		ESQL Client	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

Query/Results

Table 12–21 lists the Query/Results components and subcomponents. When you choose the Complete installation option and install Query/Results, all components and subcomponents listed are installed.

Table 12–21: Query/Results components and subcomponents

(1 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Character Base Tools	M	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Base Tools	O	ADM Runtime — CHAR	O
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	O	—	—
Character Image — Dev	O	—	—
Character Runtime Client — Dev	M	—	—
Client-Side Web Services Deploy	R	Client-Side Security	R
		Security Common	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	O	4GL Database	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M

Table 12–21: Query/Results components and subcomponents*(2 of 3)*

Component	M/R/O¹	Subcomponent	M/R/O
Graphical Base Tools	M	ADM Runtime — GUI	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
NetSetup	O	—	—
OE Build Utility	R	—	—
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
Oracle DataServer Client	O	—	—
Progress Messages (PROMSGS)	M	Language subset	O
Query Client	M	Base Client — Query	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
Remote Debugging	M	—	—
Report Engine	M	—	—
Results	M	Base ADE	M
		Results (Graphical)	M

Table 12–21: Query/Results components and subcomponents*(3 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M

1. M=Mandatory, R=Recommended, O=Optional

Translation Manager

Table 12–22 lists the Translation Manager components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed.

Table 12–22: Translation Manager components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
ActiveX Control support	M	ActiveX Control Runtime Support	M
Common Files	M	Common Files	M
		WebSpeed Common	M
NetSetup	O	–	–
Translation Manager	M	Base ADE	M
		Translation Manager	M
Visual Translator	M	–	–

1. M=Mandatory, R=Recommended, O=Optional

Visual Translator

Table 12–23 lists the Visual Translator components and subcomponents. When you choose the Complete installation option and install the Visual Translator, all components and subcomponents listed are installed.

Table 12–23: Visual Translator components and subcomponents (1 of 2)

Component	M/R/O ¹	Subcomponent	M/R/O
4GL Client	M	Base Client — 4GL	M
		Crypto Tools	M
		Graphical Client	M
		ICU PSC	M
		Java Server	M
		XML	M
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M
Graphical Base Tools	M	ADM Runtime — GUI	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Procedure Editor — GUI	M
NetSetup	O	–	–

Table 12–23: Visual Translator components and subcomponents*(2 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		DotNET Client Support	R
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	–	–
Report Engine	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
SQL Database Server	O	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		JDK	M
		Progress Databases	M
		SQL Server	M
Visual Translator	M	–	–

1. M=Mandatory, R=Recommended, O=Optional

Web Services Adapter

Table 12–24 lists the Web Services Adapter (WSA) component that is installed when you choose the Complete installation option and install the Web Services Adapter.

Table 12–24: Web Services Adapter components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Name Server	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Web Services Adapter	M	Web Services Adapter Common	M
		Web Services Adapter Inst	M

1. M=Mandatory, R=Recommended, 0=Optional

WebSpeed Messenger

Table 12–25 lists the WebSpeed Messenger components.

Table 12–25: WebSpeed Messenger components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Messenger component	M	Java Server	M
		Messenger Runtime Support	M
		WebSpeed Tools	M
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Web static	M	–	–
WebSpeed Common	M	–	–

1. M=Mandatory, R=Recommended, 0=Optional

WebSpeed Workshop

Table 12–26 lists the WebSpeed Workshop components and subcomponents. When you choose the Complete installation option and install WebSpeed Workshop, all components and subcomponents listed are installed.

Table 12–26: WebSpeed Workshop components and subcomponents

(1 of 6)

Component	M/R/O ¹	Subcomponent	M/R/O
Administration and Configuration	M	Administration Server	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Character Administration	M
		Common Broker	M
		Database Utilities	M
		Explorer Tools	M
		4GL Database	M
		Graphical Administration	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M
		WebSpeed Tools	M
Client-Side Web Services	R	Client-Side Security	R
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
4GL utilities	R	XSD-4GL	R
Common files	M	Common Files	M
		WebSpeed Common	M

Table 12–26: WebSpeed Workshop components and subcomponents*(2 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Development Data Source Option	O	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		DataDirect ODBC Driver Support	O
		ESQL Client	M
		ICU PSC	M
		JDK	M
		Progress Databases	M
		Oracle Client	O
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M
Development Server Options	R	ADM Runtime — GUI	R
		ADM Runtime — CHAR	M
		Advanced Editing	M
		APPBuilder Core	M
		AppServer — Dev	R
		Base ADE	M
		Base Client — 4GL	M
		Character Client — WebSpeed	R
		Client-Side Security	R
		Common Broker	M
		Compile Tool GUI	R
		Crypto Tools	M
		Database Server	M

Table 12–26: WebSpeed Workshop components and subcomponents*(3 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
Development Server Options (cont.)	R	Database Tools	M
		DB Administration Source	M
		Desktop	M
		Editor Source	M
		4GL Database	M
		4GL Server	M
		Graphical Client	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — CHAR	M
		Procedure Editor — GUI	M
		Progress Databases	M
		Security Common	M
		SQL Server	M
		Transaction Server — Dev	R
		WebSpeed Messenger	R
		WebSpeed Runtime	M
		WebSpeed Tools	M
		Web Static	M
		Workshop	M
		XML	M
		Progress Dynamics RT	R
Development Source Code Options	R	ADE Common Source	M
		ADM Source Code	O
		DB Administration Source	O
		Editor Source	O
		ProTools Source Code	M

Table 12–26: WebSpeed Workshop components and subcomponents*(4 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
NetSetup	O	–	–
OE Build Utility	R	–	–
Open Client Adapters Options	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		DotNET Client Support	R
		OpenEdge Adapter for Sonic ESB	R
		OpenEdge Adapter for SonicMQ	R
		Proxy Generator	M
		Web Services Adapter Common	M
		Web Services Admin Enable	R
		Web Services Schema	R
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M
Toolkit	M	–	–

Table 12–26: WebSpeed Workshop components and subcomponents*(5 of 6)*

Component	M/R/O ¹	Subcomponent	M/R/O
WebSpeed Development	R	ActiveX Control Development	M
		4GL Database	M
		ActiveX Control Runtime Support	M
		ADM Runtime CHAR	M
		ADM Runtime GUI	R
		Application Debugger	R
		Advanced Editing	M
		APPBuilder Core	M
		Base ADE	M
		Base Client — 4GL	M
		Character Client — 4GL	O
		Compile Tool GUI	R
		Crypto Tools	M
		Progress Databases	M
		Database Server	M
		Database Tools	M
		DB Administration Source	M
		Desktop	M
		Editor Source	M
		Graphical Client	M
		ICU PSC	M
		Web Static	M

Table 12–26: WebSpeed Workshop components and subcomponents (6 of 6)

Component	M/R/O ¹	Subcomponent	M/R/O
WebSpeed Development (cont.)	R	4GL Server	M
		Remote Debugging	M
		Procedure Editor — CHAR	O
		SQL Server	M
		WebSpeed Runtime	M
		WebSpeed Workshop Dev	R
		XML	M
		Procedure Editor — GUI	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Installation Products and Components on UNIX

This chapter presents a comprehensive list of the components and subcomponents that comprise each OpenEdge product. The information attempts to be a reference for all UNIX/Linux platforms to which you can install. However, depending on the specific UNIX/Linux platform, there can be some minor differences. And, in a few instances, some platform variations can affect component and subcomponent availability as noted.

Refer to the product details in this chapter when planning and performing either a Complete or Custom installation.

The topics in this chapter are described in the following sections:

- [OpenEdge Product highlights](#)
- [OpenEdge installation options](#)
- [OpenEdge product components and subcomponents](#)

Note: With the exception of Solaris SPARC 32- and 64-bit platforms, all supported UNIX/Linux platforms require at least the entry level JDK and JRE versions installed to run OpenEdge products. For more information about this Java prerequisite, see the [“Java requirements”](#) section on page 2–2.

OpenEdge Product highlights

The following sections briefly highlights some OpenEdge products.

Audit Policy Maintenance

Audit Policy Maintenance is a core business services auditing framework. Among the many functional elements of this framework is the Audit Policy Maintenance which enables you to create and maintain auditing policies for applications and databases. Application and database auditing include activities such as data collection, configuration, management, and security.

Audit Policy Maintenance is available as a subcomponent of the OpenEdge Personal RDBMS, OpenEdge Workgroup RDBMS, OpenEdge Enterprise RDBMS, Release and other OpenEdge products.

For more information about the auditing framework and the APMT, see *OpenEdge Getting Started: Core Business Services*.

OpenEdge Adapter for Sonic ESB

The OpenEdge Adapter for SonicESB, which is only available on platforms that Sonic Software supports, is used to integrate an OpenEdge application server and the Sonic Enterprise Service Bus and to create Service Oriented Architecture (SOA) applications. The adapter consists of Java JAR files and configuration files. It uses a standard OpenEdge installation procedure but does not require a complete OpenEdge environment.

The OpenEdge installation program installs a Sonic ESB container in which the OpenEdge Adapter for Sonic ESB will run. The Domain Manager for a Sonic installation requires information from this container, and the OpenEdge installation program uses the information you provide to connect to the Domain Manager and obtain this information.

Note: If the Domain Manager container is not running when you are performing the OpenEdge installation, or the Installation Utility cannot connect to the Domain Manager, a script file, either `offline.bat` or `offline.sh`, is created to hold the required information. You can run either file once the Domain Manager is running.

For more information about the OpenEdge Adapter for Sonic ESB, see *OpenEdge Development: Messaging and ESB*.

Web Services Adapter

A *Web service* is an application that can be accessed over the Internet (or an intranet) using industry-standard protocols. Web services provide an industry-standard way for all types of client applications to call functions on all types of application servers, over any network configuration that supports SOAP over HTTP, and where the application program interface (API) can be described using WSDL.

In OpenEdge, you can build and deploy Web services using the AppServer, Web service tools, and the Web Services Adapter (WSA). The Web service tools allow you to develop a Web Service-enabled AppServer application and then to deploy this application as a Web service.

The WSA is a Java servlet running on a Web server or stand-alone JSE. This Java servlet serves as the SOAP-to-business interface gateway and Web service management engine for your Web service. Thus, the WSA provides user access to the Web service WSDL file and supports Web service administration; it also manages the exchange of Web service SOAP requests and generates SOAP responses between Web service clients and AppServers™ at run time.

You can install the WSA as a separate stand-alone product on its own machine. It is available as a separately installable OpenEdge component, as a sub-component of the OpenEdge AppServer Enterprise product and of the OpenEdge Development Server product. It is also available as a download from the Progress Download Center Web site at <http://www.progress.com/esd>. For information on WSA configuration, see the WSA sections of *OpenEdge Development: Web Services* and *OpenEdge Application Server: Administration*.

OpenEdge installation options

You can choose between two options when installing OpenEdge: complete or custom. These two options allow you to choose the option that is best for you, depending on how many products you are installing, which product components are mandatory and which are optional, and whether all the products reside on the same system.

Complete installation option

When you choose the Complete installation option and specify the products you want to install, all mandatory, recommended, and optional components and subcomponents are installed automatically. For this reason, a Complete installation usually requires more disk space than a custom installation requires.

Custom installation option

When you choose the custom installation option, all mandatory products and subcomponents are installed, but you can selectively install the recommended and optional components and subcomponents on a product-by-product basis. A custom installation provides more advanced users, for whom this method is recommended, a means to distribute OpenEdge components on different machines, select product components to suit their business needs, and work around issues such as disk space limitations.

Caution: Removing recommended product components and/or subcomponents can affect the functionality of a product.

The mandatory, recommended, and optional components and subcomponents for each OpenEdge product are listed, by product, in the tables that follow.

OpenEdge product components and subcomponents

The following tables list the components and subcomponents that are installed for each product.

4GL Development System

[Table 13–1](#) lists the 4GL Development System components and subcomponents that are installed when you choose the Complete installation option and install the 4GL Development System.

Table 13–1: 4GL Development System components and subcomponents

(1 of 5)

Component	M/R/O ¹	Subcomponent	M/R/O
4GL Client	M	Base Client — GL	M
		Character Client	M
		Crypto Tools	M
		ICU PSC	M
		Java Server	M
		XML	M
4GL utilities	R	XSD — 4GL	R
ADE Source Code	R	ADE Common Source	M
		ADM Source	M
		DB Administration Source	M
		Editor Source	M
Application Debugger	R	Application Debugger	R
		Remote Debugging	M
Character Base Tools	M	ADM Runtime — CHAR	M
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M

Table 13–1: 4GL Development System components and subcomponents*(2 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Client-Side Web Services	R	Client-side security	R
		Security Common ²	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	M	4GL Database	M
		Audit Policy Maintenance	M
		Base ADE	M
		Character Administration	M
		Database Utilities	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M
Name Server	M	–	–
OE Build Utility	R	–	–

Table 13–1: 4GL Development System components and subcomponents*(3 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
Open Client Adapter Options	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for SonicMQ	R
		OpenEdge Adapter for Sonic ESB	R
		Proxy Generator	M
		Web Services Adapter Common	M
		Web Services Admin Enabler	R
		Java Class Tailoring	M
		Web Services Schema	R
Open Edge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL Server	M

Table 13–1: 4GL Development System components and subcomponents*(4 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL ODBC Client	M
		SQL Server	M
Oracle DataServer Client	O	–	–
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M

Table 13–1: 4GL Development System components and subcomponents*(5 of 5)*

Component	M/R/O ¹	Subcomponent	M/R/O
SQL Database Server	O	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		JDK	M
		Progress Databases	M
		SQL Server	M
Toolkit	M	–	–

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

AppServer Internet Adapter (AIA)

Table 13–2 lists the AppServer Internet Adapter components that are installed when you choose the Complete installation option and install the AppServer Internet Adapter.

Table 13–2: AppServer Internet Adapter components and subcomponents*(1 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Internet Adapter	M	–	–
Common Broker	M	–	–
Common Files (minimum)	M	–	–
Java Server	M	Java Server	M
Name Server	M	–	–
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M

Table 13–2: AppServer Internet Adapter components and subcomponents*(2 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

Client Networking

[Table 13–3](#) lists the Client Networking components and subcomponents. When you choose the Complete installation option and install Client Networking, all components and subcomponents listed are installed.

Table 13–3: Client Networking components and subcomponents*(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Character Base Tools	M	ADM Runtime — CHAR	M
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–3: Client Networking components and subcomponents*(2 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Database Administration Tools	M	4GL Database	M
		Audit Policy Maintenance	M
		Base ADE	M
		Character Administration	M
		Database Utilities	M
Name Server	M	—	—
OE Build Utility	R	—	—
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		Open Edge Adapter for SonicMQ	R
OpenEdge ESQ/C Clients	O	Database Tools	M
		ESQ/C Client	M
		ICU PSC	M
		SQL Common	M
		SQL Server	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL Server	M

Table 13–3: Client Networking components and subcomponents*(3 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL Common	M
		SQL ODBC Client	M
		SQL Server	M
Oracle DataServer Client	O	—	—
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Runtime Client	M	Base Client — RT	M
		Character Client	M
		Crypto Tools	M
		ICU PSC	M
		Java Server	M
		XML	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M

1. M=Mandatory;R=Recommended;O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Replication

Table 13–4 lists the OpenEdge Replication components and subcomponents that are installed when you choose the Complete Installation option.

Table 13–4: OpenEdge Replication components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Progress Messages (PROMSGS)	M	Language subset	O
Replication	M	Replication Common	M
		Replication Installation	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Replication Plus

Table 13–5 lists the OpenEdge Replication Plus components and subcomponents that are installed when you choose the Complete Installation option.

Table 13–5: OpenEdge Replication Plus components and subcomponents

(1 of 2)

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Progress Messages (PROMSGS)	M	Language subset	O

Table 13–5: OpenEdge Replication Plus components and subcomponents (2 of 2)

Component	M/R/O ¹	Subcomponent	M/R/O
Replication	M	Replication Common	M
		Replication Installation	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M

- 1. M=Mandatory, R=Recommended, O=Optional
- 2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

NameServer

Table 13–6 lists the NameServer components and subcomponents that are installed when you choose the Complete Installation option.

Table 13–6: NameServer components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
NameServer	M	—	—
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	Language subset	O

- 1. M=Mandatory, R=Recommended, O=Optional

NameServer Load Balancer

Table 13–7 lists the NameServer Load Balancer components and subcomponents that are installed when you choose the Complete installation option and install the NameServer Load Balancer.

Table 13–7: NameServer Load Balancer components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Adapter for SonicESB

Table 13–8 lists the OpenEdge Adapter for SonicESB components and subcomponents. that are installed when you choose the Complete installation option.

Table 13–8: SonicMQ Adapter components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Java Class Tailoring	M	–	–
OpenEdge Adapter for Sonic ESB	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Application Server — Basic

Table 13–9 lists the OpenEdge Application Server Basic components and subcomponents that are installed when you choose the Complete installation option and install the Application Server Basic option.

Table 13–9: OpenEdge Application Server — Basic components and subcomponents *(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Runtime Client	M	–	–
Basic Server Options	R	SQL Server	M
		AppServer — Basic	R
		ADE Common Source	M
		ADM Runtime CHAR	M
		Audit Policy Maintenance	M
		Base ADE	M
		Base Client — 4GL	M
		Character Client	M
		Common Broker	M
		Crypto Tools	M
		ICU PSC	M
		Name Server	R
		Procedure Editor CHAR	M
		Progress Databases	M
		Transaction Server — Basic	R
		Editor Source	M
		Web Static	M
		WebSpeed Messenger	R
		WebSpeed Runtime	M
		XML	M

Table 13–9: OpenEdge Application Server — Basic components and subcomponents (2 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Schema	R
		Web Services Basic	R
Common Files	M	Common Files	M
		WebSpeed Common	M
OE Build Utility	R	—	—
OE Perl	M	—	—
OpenClient Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Open Edge Adapter for Sonic MQ	R
		Java Client Support	R
		Java Server	M
		Java Ext	M
		Java Class Tailoring	M
Progress Messages (PROMSGS)	M	All languages	O
Remote Debugging	M	—	—
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common	M
		Server-Side Security	M

Table 13–9: OpenEdge Application Server — Basic components and subcomponents (3 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Server Admin and Configuration	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M
Server Data Source Options	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		Oracle Client	O
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Application Server — Enterprise

[Table 13–10](#) lists the OpenEdge Application Server Enterprise components and subcomponents that are installed when you choose the Complete option.

Table 13–10: Application Server — Enterprise components and subcomponents (1 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
AppServer Runtime Client	M	—	—
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R

Table 13–10: Application Server — Enterprise components and subcomponents (2 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
Client-side Web Services Deploy (cont.)	R	Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Enterprise Server Options	R	AppServer — Enterprise	R
		ADE Common Source	M
		ADM Runtime CHAR	M
		Base ADE	M
		Base Client — 4GL	M
		Character Client	M
		Client-side Security	R
		Common Broker	M
		Crypto Tools	M
		Editor Source	M
		ICU PSC	M
		Auditing Policy Maintenance	M
		NameServer	R
		Procedure Editor CHAR	M
		Progress Databases	M
		Security Common ²	M
		SQL Server	M
		Transaction Server — Enterprise	R
		Web Static	M
		WebSpeed Messenger	R
		WebSpeed Run-time	M
		XML	M
OE Build Utilities	R	—	—
OE Perl	M	—	—

Table 13–10: Application Server — Enterprise components and subcomponents *(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Progress Messages (PROMSGS)	M	All Languages	O
Remote Debugging	M	—	—
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M
Open Client Adapter Options Enterprise	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for SonicMQ	R
		OpenEdge Adapter for Sonic ESB	R
		Web Services Adapter	M
		Web Services Admin Enabler	R
		Java Class Tailoring	M
		Web Services Schema	R
Server Admin and Configuration	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M

Table 13–10: Application Server — Enterprise components and subcomponents *(4 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Server Data Source Options	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		Oracle Client	O
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge DataServer for Oracle

[Table 13–11](#) lists the Oracle DataServer components and subcomponents that are installed when you choose the Complete installation option and install the OpenEdge DataServer for Oracle.

Table 13–11: OpenEdge DataServer for Oracle components and subcomponents *(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Character Base Tools	M	ADM Runtime — CHAR	M
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–11: OpenEdge DataServer for Oracle components and subcomponents (2 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Database Administration Tools	M	4GL Database	M
		Audit Policy Maintenance Tool	M
		Base ADE	M
		Character Administration	M
		Database Utilities	M
Name Server	M	–	–
OE Build Utility	R	–	–
OpenClient Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for SonicMQ	R
		Java Class Tailoring	M
Oracle DataServer	M	Broker	M
		Oracle DataServer	M
Oracle DataServer Client	O	–	–
Progress Explorer Tools	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
		Ubroker Tools	M

Table 13–11: OpenEdge DataServer for Oracle components and subcomponents (3 of 3)

Component	M/R/O ¹	Subcomponent	M/R/O
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	–	–
Runtime Client	M	Base Client — RT	M
		Character Client	M
		Crypto Tools	M
		ICU PSC	M
		Java Server	M
		XML	M
Schema Holder and Server	M	4GL Server	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

OpenEdge Development Server

Table 13–12 lists the OpenEdge Development Server components and subcomponents. When you choose the Complete installation option and install the OpenEdge Development Server, all components and subcomponents listed are installed.

Table 13–12: OpenEdge Development Server components and subcomponents *(1 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Administration and Configuration	M	4GL Database	M
		Administration Server	M
		Auditing Policy Maintenance	M
		Base ADE	M
		Character Administration	M
		Common Broker	M
		Database Utilities	M
		Java Ext	M
		Java Server	M
		Name Server	R
		Ubroker Tools	M
Client-Side Web Services	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–12: OpenEdge Development Server components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Development Server Options	R	4GL Database	M
		4GL Server	M
		ADM Runtime — CHAR	M
		AppServer — Development	R
		Base ADE	M
		Base Client — 4GL	M
		Character Client	M
		Client-side Security	R
		Common Broker	M
		Crypto Tools	M
		Database Server	M
		Database Tools	M
		ICU PSC	M
		NameServer	R
		Procedure Editor — CHAR	M
		Progress Databases	M
		Security Common ²	M
		SQL Server	M
		WebSpeed Messenger	R
		Transaction Server — Development	M
		Web Static	M
		WebSpeed Messenger	R
		WebSpeed Runtime	M
		XML	M
		Editor Source	M
		Progress Dynamics RT	R

Table 13–12: OpenEdge Development Server components and subcomponents *(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Development Data Source Option	O	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ESQL Client	M
		ICU PSC	M
		JDK	M
		Oracle Client	O
		Progress Databases	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M
Open Client Adapter Options	R	Web Services Schema	R
		AppServer Internet Adapter	R
		Common Broker	M
		Java Class Tailoring	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		OpenEdge Adapter for Sonic ESB	R
		OpenEdge Adapter for SonicMQ	R
		Proxy Generator	M
		Web Services Adapter Comm	M
		Web Services Admin Enable	R

Table 13–12: OpenEdge Development Server components and subcomponents (4 of 4)

Component	M/R/O ¹	Subcomponent	M/R/O
OE Build Utility	R	–	–
OE Perl	M	–	–
Progress Messages (PROMSGS)	M	Language Subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M
Server Source Code Options	R	ADE Common Source	O
		Editor Source	O
Toolkit	M	–	–
Other Development Server Options	R	ADM Runtime — CHAR	O
		Application Debugger	R
		Character Client — Runtime	O
		Compile Tool — CHAR	O
		Crypto Tools	M
		Procedure Editor — CHAR	O
		Remote Debugging	M
4GL Utilities	R	XSD — 4GL	R

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Enterprise RDBMS

Table 13–13 lists the Enterprise RDBMS components and subcomponents that are installed when you choose the Complete installation option and install the OpenEdge Enterprise RDBMS.

Table 13–13: OpenEdge Enterprise RDBMS components and subcomponents *(1 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
4GL	O	4GL Server	M
		ADM Runtime — CHAR	O
		Audit Policy Maintenance	M
		Base ADE	M
		Base Client — DA	M
		Character Administration	M
		Character Client	M
		Compile Tool — CHAR	O
		Crypto Tools	M
		ICU PSC	M
		Oracle Client	O
		Procedure Editor — CHAR	M
		SQL Server	M
		XML	M
		Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
Client-side Web Services Deploy	R	Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–13: OpenEdge Enterprise RDBMS components and subcomponents*(2 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
Database Management	M	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
		Progress Databases	M
		SQL Server	M
Failover Clusters	R	Cluster Common	M
Name Server	M	–	–
OE Build Utility	R	–	–
OE Perl	M	–	–
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M

Table 13–13: OpenEdge Enterprise RDBMS components and subcomponents*(3 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL JDBC Client	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL ODBC Client	M
		SQL Server	M
		SQL Common	M
Progress Explorer Tools — DB	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M

Table 13–13: OpenEdge Enterprise RDBMS components and subcomponents*(4 of 4)*

Component	M/R/O ¹	Subcomponent	M/R/O
SQL	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Personal RDBMS

Table 13–14 lists the Personal RDBMS components and subcomponents that are installed when you choose the Complete installation option and install the OpenEdge Personal RDBMS.

Table 13–14: OpenEdge Personal RDBMS components and subcomponents *(1 of 3)*

Component	M/R/O	Subcomponent	M/R/O
4GL ¹	O	4GL Server	M
		ADM Runtime — CHAR	O
		Audit Policy Maintenance	M
		Base ADE	M
		Base Client — DA	M
		Character Administration	M
		Character Client	M
		Compile Tool — CHAR	O
		Crypto Tools	M
		ICU PSC	M
		Oracle Client	O
		Procedure Editor — CHAR	M
		SQL Server	M
		XML	M
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–14: OpenEdge Personal RDBMS components and subcomponents*(2 of 3)*

Component	M/R/O	Subcomponent	M/R/O
Database Management	M	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
		Progress Databases	M
		SQL Server	M
Name Server	M	–	–
OE Build Utility	R	–	–
OE Perl	M	–	–
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M

Table 13–14: OpenEdge Personal RDBMS components and subcomponents*(3 of 3)*

Component	M/R/O	Subcomponent	M/R/O
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL JDBC Client	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL ODBC Client	M
		SQL Server	M
		SQL Common	M
Progress Explorer Tools — DB	M	Administration Server	M
		Common Broker	M
		Java Ext	M
		Java Server	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M
SQL	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. The 4GL component of the OpenEdge Personal RDBMS includes the Client Networking functionality.

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge Workgroup RDBMS

Table 13–15 lists the Workgroup RDBMS components and subcomponents that are installed when you choose the Complete installation option and install the OpenEdge Workgroup RDBMS.

Table 13–15: OpenEdge Workgroup RDBMS components and subcomponents *(1 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
4GL	O	4GL Server	M
		ADM Run-time — CHAR	O
		Auditing Policy Maintenance	M
		Base ADE	M
		Base Client — DA	M
		Character Administration	M
		Character Client	M
		Compile Tool — CHAR	O
		Crypto Tools	M
		ICU PSC	M
		Oracle Client	O
		Procedure Editor — CHAR	M
		SQL Server	M
		XML	M
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
		WSDL Analyzer	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M

Table 13–15: OpenEdge Workgroup RDBMS components and subcomponents*(2 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
Database Management	M	4GL Server	M
		Database Server	M
		Database Tools	M
		Database Utilities	M
		ICU PSC	M
		Legacy 83 Utilities	M
		Legacy 91 Utilities	M
		OE 10.1B DB Utilities	M
		Progress Databases	M
		SQL Server	M
Name Server	M	–	–
OE Build Utility	R	–	–
OE Perl	M	–	–
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Monitoring	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R
OE ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M

Table 13–15: OpenEdge Workgroup RDBMS components and subcomponents*(3 of 3)*

Component	M/R/O ¹	Subcomponent	M/R/O
OE SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL JDBC Client	M
		SQL Common	M
		SQL Server	M
OE SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL ODBC Client	M
		SQL Common	M
Progress Explorer Tools — DB	M	Administration Server	M
		Java Ext	M
		Java Server	M
		Common Broker	M
Progress Messages (PROMSGS)	M	Language subset	O
Remote Debugging	M	—	—
Secure Server	M	Perl	M
		Security Common ²	M
		Server-Side Security	M
SQL	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Common	M
		SQL JDBC Client	M
		SQL ODBC Client	M
		SQL Server	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

OpenEdge SQL Client Access

Table 13–16 lists the SQL Client Access components and subcomponents that are installed when you choose the Complete installation option and install Client Access.

Table 13–16: OpenEdge SQL Client Access components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Common Files	M	Common Files	M
		WebSpeed Common	M
Name Server	M	–	–
OpenEdge ESQL/C Clients	O	Database Tools	M
		ESQL Client	M
		ICU PSC	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL JDBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL JDBC Client	M
		SQL Server	M
		SQL Common	M
OpenEdge SQL ODBC Clients	O	Database Tools	M
		ICU PSC	M
		SQL ODBC Client	M
		SQL Server	M
		SQL Common	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on the Linux Power PC platform.

Query/Results

Table 13–17 lists the Query/Results components and subcomponents that are installed when you choose the Complete installation option and install Query/Results.

Table 13–17: Query/Results components and subcomponents

(1 of 2)

Component	M/R/O ¹	Subcomponent	M/R/O
Character Base Tools	M	ADM Runtime — CHAR	M
		Base ADE	M
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Client-side Web Services Deploy	R	Client-side Security	R
		Security Common ²	M
		Web Services Basic	R
		Web Services Schema	R
Common Files	M	Common Files	M
		WebSpeed Common	M
Database Administration Tools	O	4GL Database	M
		Audit Policy Maintenance	M
		Base ADE	M
		Character Administration	M
		Database Utilities	M
OE Build Utility	R	—	—
Open Client Adapter Options Basic	R	AppServer Internet Adapter	R
		Common Broker	M
		Java Client Support	R
		Java Ext	M
		Java Server	M
		Java Class Tailoring	M
		OpenEdge Adapter for SonicMQ	R

Table 13–17: Query/Results components and subcomponents*(2 of 2)*

Component	M/R/O ¹	Subcomponent	M/R/O
Oracle DataServer Client	O	–	–
Progress Messages (PROMSGS)	M	Language Subset	O
Query Client	M	Base Client — Query	M
		Character Client	M
		Crypto Tools	M
		ICU PSC	M
		Java Server	M
		XML	M
Remote Debugging	M	–	–
Results	M	Base ADE	M
		Results (Char)	M
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M

1. M=Mandatory, R=Recommended, O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

WebSpeed Messenger

Table 13–18 lists the WebSpeed Messenger components that are installed when you choose the Complete option.

Table 13–18: WebSpeed Messenger components

Component	M/R/O ¹	Subcomponent	M/R/O
Messenger Component	M	Java Server	M
		Messenger Runtime Support	M
Progress Messages (PROMSGS)	M	Language subset	O
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Web Static	M	–	–
WebSpeed Common	M	–	–

1. M=Mandatory;R=Recommended;O=Optional

2. The Security Common subcomponent is not supported on a Linux PowerPC platform.

Web Services Adapter

Table 13–19 lists the Web Services Adapter components that are installed when you choose the Complete installation option.

Table 13–19: Web Services Adapter components and subcomponents

Component	M/R/O ¹	Subcomponent	M/R/O
Name Server	M	–	–
Secure Clients	M	Client-Side Security	R
		Perl	M
		Security Common ²	M
Web Services Adapter	M	Web Services Adapter Common	M
		Web Services Adapter Install	M

- 1. M=Mandatory;R=Recommended;O=Optional
- 2. The Security Common subcomponent is not supported on the Linux Power PC platform.

Preinstallation Checklist for Windows

The checklist is a tool to help technical personnel determine and record product installation choices **before** running the OpenEdge Release 10.1B Installation Utility. Familiarize yourself with the checklist's content and then work through each topic. Refer to this completed checklist when you start the installation.

Before you start

Obtain these documents to complete the checklist:

- ☐ *License Addendum* and *OpenEdge Release Notes* from the OpenEdge package.
- ☐ Windows Installation online help available for download from the root directory of your installation media (CD or DVD), or from the .tar file from the Progress Download Center at <http://www.progress.com/esd>.
- ☐ “Chapter 12, “OpenEdge Installation Products and Components in Windows” in *OpenEdge Getting Started: Installation and Configuration*.

Products to install

- ☐ Obtain serial numbers and control codes from the *License Addendum*. Control codes are not case sensitive and can be entered in any order.

Prerequisite third-party software

If your planned installation meets each of the following conditions, you are prompted to accept a Microsoft .NET Framework software installation which will run after the OpenEdge installation concludes:

- ☐ At least one product for which you enter a control code requires Microsoft .NET Framework software. The OpenEdge 10.1B development products that require .NET Framework are:

4GL Development System	OpenEdgeArchitect
OpenEdge Studio	WebSpeedWorkshop

- ☐ The Microsoft .NET Framework is not currently installed on the machine to which you are installing OpenEdge.

Values from your existing OpenEdge installation

If the Installation Program detects an OpenEdge 10.1A installation on the machine to which you are installing OpenEdge 10.1B, you are prompted to use the field values defined for the 10.1A installation as default field values in the OpenEdge 10.1B installation. Proceed as follows:

- ☐ Choose **Yes** to accept the values as default field values. A reminder **Initial values used from a previous installation** displays at the bottom of each dialog box affected.
- ☐ Choose **No** to decline the option. Only the Installation Program default field values appear.

Installation and working directories

- ☐ Accept the default installation directory location provided, **C:\Progress\OpenEdge**, or define another location: _____. **Do not** use the pathname where other versions of Progress or OpenEdge products are already installed on your system.
- ☐ Accept the default working directory location provided, **C:\OpenEdge\WRK**, in which your applications, databases, and log files will reside, or define another location: _____. **Do not** make your working directory a subdirectory under the OpenEdge installation path.

Installation type

- ☐ Complete (default option)
- ☐ Custom

Recommended and optional components

Depending on the products for which you enter control codes, you can be prompted to identify the following components for the purposes indicated:

OpenEdge Adapter for Sonic ESB and/or WebSpeed with local Web Server

Each is a recommended component that must be configured during the installation.

For example, the OpenEdge Adapter for Sonic ESB is a recommended component of OpenEdge Application Server Enterprise Edition and OpenEdge Studio. Many OpenEdge products required access to at least one of the recognized Web server types. See the “[Web server](#)” section on page A–6 for a list of products.

Proceed with this choice based on these options:

- ☐ Accept the check mark, or select the check box associated with a recommended component to identify it for configuration; during the installation, you will be prompted to specify the component’s configuration details.

To note configuration details for the OpenEdge Adapter for Sonic ESB in the “[OpenEdge Adapter for Sonic ESB](#)” section on page A–5. To note configuration details for the Web Server type, see the “[Web server](#)” section on page A–6.

- ☐ Deselect the check box if you plan not to install the recommended component or leave the check box blank. (If a check box is grayed out, the component is not available.)

Progress Dynamics

An optional component that can be installed with OpenEdge products such as OpenEdge Studio and OpenEdge Architect. When performing a Complete installation, determine how to proceed with this choice based on these options:

- ☐ Accept the check mark associated with Progress Dynamics when installing OpenEdge Studio, or deselect it to bypass installing the optional component.
- ☐ If you are installing OpenEdge Architect, the Progress Dynamics component is **unchecked** by default. The Progress Dynamics component supports the AppBuilder functionality within OpenEdge Architect. You must select this option on this dialog box to install it so that you can enable Progress Dynamics in OpenEdge Architect.

To note configuration details for the Progress Dynamics option in the “[Progress Dynamics](#)” section on page A–7.

Database

If you are installing a database, the database server connection to support queries written in ABL is installed by default:

- ☐ Accept the system-default to install the SQL database server connection to support queries written in SQL, or deselect the option.

OpenEdge Adapter for Sonic ESB

Respond to the following points if you are configuring the OpenEdge Adapter for Sonic ESB as a recommended component or as a stand-alone product (downloadable from the Progress Download Center):

- ☐ Accept the default host name value provided in the **Container Name** field **or** define a different container name; it must be unique to the Sonic management broker:
_____.
- ☐ Accept the default values provided in the **Domain Name**, **Connection URL**, **User Name** and **Password** fields, **or** define unique values for any of these fields:
_____.
- ☐ The OpenEdge Adapter for Sonic ESB requires the configuration of the Sonic Management Console to administer OpenEdge Services. Proceed as shown in the following table:

If the Sonic Domain Manager is installed . . .	Then . . .
On a machine on which you are installing all other OpenEdge products.	During the installation, select the local directory in which the Domain Manager resides. Specify your choice in the Sonic ESB Home Directory field: _____.
On a machine remote from all other OpenEdge products you are installing.	As a postinstallation task, you must manually configure the remote system’s Sonic Management Console, using the installation instructions available in <i>OpenEdge Application Server: Administration</i> .

Web server

You must have access to at least one of the recognized Web server types if you plan to install any of the following OpenEdge products:

- The WebSpeed Messenger, Secure AppServer Internet Adapter (AIA) or Web Services Adapter (WSA) products to develop and/or deploy Web applications.
- OpenEdge Development Server and you plan to use Progress Dynamics functionality.
- WebSpeed as a component or subcomponent of another OpenEdge product, and you plan to use this functionality. (If you do not plan to use WebSpeed functionality and the check box is available, leave the check box blank to bypass these Web server choices during the installation process.)

If you are configuring a web server, respond to the following points for each Web server you intend to use. Refer to the Installation online help topic “Selecting a Web Server Type,” as needed:

- ☐ Identify the type of Web server:
 - Microsoft Internet Information System (IIS) or ISAPI-compatible.
 - Sun Web server or NSAPI-compatible.
 - CGI-compatible.
- ☐ Note the Web server’s machine location:
 - On the same machine where you plan to install your OpenEdge products.
 - On a machine that is different from the machine on which you are installing your OpenEdge products.
- ☐ For a Microsoft Internet Information System (IIS) or ISAPI-compatible Web server:
 - Accept the default directories provided that appear in the **Web Server Script directory** and the separate **Web Server Document Root directory fields**, respectively. (These default directories are independent of each other.) **Or**, you can select a different location for either or both directories:
_____. (If you select an alternative Web Server script directory, it must be an existing directory that your Web server references.)

-
- Deselect the **Copy static HTML files to Document Root directory** checkbox and select the **Create virtual directory for static HTML files** checkbox. This step enables OpenEdge to create an alias that points to the WebSpeed HTML files in the OpenEdge installation directory.
- ☐ For either a Sun Web server or NSAPI-compatible Web server **or** a CGI-compatible Web server:
- Define the directory path for the **Web Server Scripts directory** field:
_____ and the **Web Server Document Root directory** field: _____.
 - Select the Web server's document root directory for the **Copy static HTML to the Document Root directory** field so that during the installation, the WebSpeed Workshop HTML files are copied to this location: _____.

Progress Dynamics

Respond to the following points, preparing the values you enter in the **Dynamics Options** dialog box:

- ☐ Select the **Install/upgrade Dynamics repository** check box to create or upgrade the icfdb repository file in the default location **C:\OpenEdge\WRK\databases**, **or** select another directory location: _____.
- ☐ If you plan to develop Web applications with this product, proceed with **one** of these tasks:
- Select the **Copy the Progress Dynamics static HTML files to the Document Root Directory** option to copy the static Dynamic Web files to your Web server's document root directory.
 - Create a virtual directory on your Web server that points to the location of the static files. (The static files physically reside in `install\tty\icf\ry`.)

Language in which online messages appear

The Progress Messages file PROMSGS contains error and information messages that appear when you are working in OpenEdge. Respond to these points related to PROMSGS:

- ☐ Identify the default language in which error and information messages appear: _____.
- ☐ Identify other languages to install (optional): _____.

For a complete list of languages shipped with OpenEdge 10.1B and supplemental PROMSGS translations that are available for download from the Progress Download Center at <http://www.progress.com/esd>, see Appendix D, “OpenEdge National Language Support,” in *OpenEdge Getting Started: Installation and Configuration*.

Character set, date, and number formats

- ☐ Select the format in which the character set, date, and numbers appear from the lists associated with each option:
 - **Character Set** (including **Collation** and **Case**): _____.
 - **Date Format**: _____.
 - **Number Format**: _____.

The software uses the settings you choose to tailor your OpenEdge startup parameter file, startup.pf.

Web Services Adapter (WSA)

If you are installing the Web Services Adapter (WSA) product or a product that contains the WSA, respond to these points:

- ☐ Accept the default URL provided in the **URL** field, **http://<machinename:80>/wsa/wsa1**, or change it to _____.

Consider your WSA configuration and your Web Server or Java Servlet engine. The URL field defines the location for the sample Web Services wsa1. (When you deploy a Web service, you deploy it to a WSA instance, which defines the root URL used to access the Web service and handles all of its client communications. Each WSA instance manages its own set of deployed Web services.)

- ☐ Retain or change the WSA authentication feature.

The default security setting to perform administrative tasks requires a valid username and password each time a connection to a WSA instance is initiated. If you select the check box associated with the **Disable Authentication** option, you disable this default security setting, eliminating any authorization requirement to administer the WSA.

Options to secure your AdminServer

Set up AdminServer authorization options either during **or** after the installation process.

- ☐ **During the installation process** — You can optionally set up both, one, or none of these AdminServer authorization options:

This AdminServer security option . . .	Requires you to . . .
User (individual) Authorization option	<ul style="list-style-type: none">• Select in the Require a Username and Password checkbox.• Set up a valid individual user name and password at the operating system level for each individual to whom you are granting individual privileges.
Group¹ Authorization option	<ul style="list-style-type: none">• Select in the Enable Group Checking checkbox.• Accept the initial default group provided, PSC Admin, which you can then use as a template for each unique group that you define.

1. For details about the guidelines, naming conventions, and restrictions concerning the **Group Authorization** option, refer to the Installation online help topic “Establishing AdminServer Authorization Options.”

- ☐ **As a postinstallation task** — To create groups or create additional groups, you will need to reinstall your OpenEdge medium and perform the **Group Authorization** option tasks at that time.

Preinstallation Checklist for UNIX

The checklist is a tool to help technical personnel determine and record product installation choices **before** running the OpenEdge Release 10.1B Installation Utility. Familiarize yourself with the checklist's content and then work through each topic. Refer to this completed checklist when you start the installation.

Before you start

Obtain these documents to complete the checklist:

- ☐ *License Addendum* and *OpenEdge Release Notes* from the OpenEdge package.
- ☐ UNIX Installation online help available for download from the root directory of your installation media (CD or DVD) or the .tar file from the Progress Download Center at <http://www.progress.com/esd>.
- ☐ Chapter 13, “OpenEdge Installation Products and Components on UNIX” in *OpenEdge Getting Started: Installation and Configuration*.

Java platform requirements

If your installation requires either the JDK or the JRE, proceed according to these points:

- ☐ Check the Java-specific requirements for your supported platform as identified in the “Chapter 2, “UNIX Systems Installation Requirements” in *OpenEdge Getting Started: Installation and Configuration*.
- ☐ If the required JDK or JRE is not provided with your installation medium, install the required element, ensuring that the JDK or JRE is the first Java entry in the \$PATH environment variable. The variable’s value must point to `<java install dir>/bin`.

Products to install

- ☐ Obtain serial numbers and control codes from the *License Addendum*. Control codes are not case sensitive and can be entered in any order.

Values from your existing OpenEdge installation

If the Installation Program detects an OpenEdge 10.1A installation on the machine to which you are installing OpenEdge 10.1B, you are prompted to use the values defined for the 10.1A installation as default values in the OpenEdge 10.1B installation. Proceed as follows:

- ☐ Choose **Yes** to accept the values as default values. A reminder **Initial values used from a previous installation** displays at the bottom of each dialog box affected.
- ☐ Choose **No** to decline the option. Only the Installation Program default values appear.

Installation and working directories

- ☐ Accept the default destination directory location provided, **/usr/OpenEdge-install-dir**, or define another location: _____. **Do not** use the pathname where other versions of Progress or OpenEdge products are already installed on your system.
- ☐ Accept the default working directory location provided, **/usr/wrk**, in which your applications, databases, and log files will reside, or define another location: _____. **Do not** make your working directory a subdirectory under the OpenEdge installation path.

Installation type

- ☐ Complete (default option)
- ☐ Custom

Database

If you are installing a database, the database server connection to support queries written in ABL is installed by default:

- ☐ Accept the system-default to install the SQL database server connection to support queries written in SQL, or deselect the option.

OpenEdge Adapter for Sonic ESB

Respond to the following points if you are installing the OpenEdgeAdapter for Sonic ESB in either of the following:

- As a stand-alone product (downloadable from the Progress Download Center).
- As a component of another product, such as the OpenEdge Application Server Enterprise Edition and 4GL Development System, and you want to install the adapter. (If you do not want to install the adapter, choose **No** to bypass the adapter choices during the installation process.)

- ☐ Accept the host name default value provided for the **Container Name** field **or** define a different container name; it must be unique to the Sonic management broker:
_____.
- ☐ Accept the default values provided for the **Domain Name**, **Connection URL**, **User Name** and **Password** fields, **or** define unique values for any of these fields:
_____.
- ☐ The OpenEdge Adapter for Sonic ESB requires the configuration of the Sonic Management Console to administer OpenEdge Services. Proceed as shown in the following table:

If the Sonic Domain Manager is installed . . .	Then . . .
On a machine on which you are installing all other OpenEdge products.	During the installation, select the local directory in which the Domain Manager resides. Specify your choice in the SonicESB Home Directory field: _____.
On a machine remote from all other OpenEdge products you are installing.	As a postinstallation task, you must manually configure the remote system's Sonic Management Console, using the installation instructions available in <i>OpenEdge Application Server: Administration</i> .

Web server

You must have access to at least one of the recognized Web server types if you plan to install any of the following OpenEdge products:

- The WebSpeed Messenger, Secure AppServer Internet Adapter (AIA), or Web Services Adapter (WSA) products to develop and/or deploy Web applications.
- WebSpeed as a component or subcomponent of another OpenEdge product, and you plan to use this functionality. (If you do not plan to use WebSpeed functionality, choose **No** to bypass these Web server choices during the installation process.)

Respond to the following points for each Web server you intend to use, referring to the Installation online help topic “Selecting a Web Server Type,” as needed:

- ☐ Identify Web server type:
 - Sun Web server or NSAPI-compatible Web server.
 - CGI-compatible Web server.
- ☐ Identify the Web server’s location:
 - On the same machine where you plan to install your OpenEdge products.
 - On a machine that is different from the machine on which you are installing your OpenEdge products.
- ☐ For either a Sun Web server or NSAPI-compatible Web server **or** CGI-compatible Web server:
 - Identify the directory you will define as your default value for the **Web Server Script directory** field: _____.
 - Identify the Web server’s document root directory you will define for the **Copy static HTML to the docroot** directory field to enable the WebSpeed Workshop HTML files to be copied to this location during the installation:
_____.

Language in which messages appear

The Progress Messages file PROMSGS contains error and information messages that appear when you are working in OpenEdge. Respond to these points related to PROMSGS:

- ☐ Identify the default language in which error and information messages appear:
_____.
- ☐ Identify other languages to install (optional): _____.

For a complete list of languages shipped with OpenEdge 10.1B and supplemental PROMSGS translations available for download from the Progress Download Center at <http://www.progress.com/esd>, see Appendix D, “OpenEdge National Language Support,” in *OpenEdge Getting Started: Installation and Configuration*.

Character set, date, and number formats

- ☐ Select the format in which the character set, date, and numbers appear from the lists associated with each option:

- **Character Set** (including **Collation** and **Case**): _____.
- **Date Format**: _____.
- **Number Format**: _____.

The software uses the settings you choose to tailor your OpenEdge startup parameter file, `startup.pf`.

Web Services Adapter (WSA)

If you are installing the WSA or a product that contains the WSA:

- ☐ Accept the default URL provided in the **URL** field, **http://<machinename:80>/wsa/wsa1**, or change it to _____.

Consider your WSA configuration and your Web Server or Java Servlet engine. The URL field defines the location of the sample Web Services `wsa1`. (When you deploy a Web service, you deploy it to a WSA instance, which defines the root URL used to access the Web service and handles all of its client communications. Each WSA instance manages its own set of deployed Web services.)

- ☐ Type **Y** to remove the default authorization requirement **or** type **N** to retain the default authorization requirement.

For the WSA authentication option, the default security setting to perform administrative tasks requires a valid username and password each time a connection to a WSA instance is initiated.

OpenEdge product scripts and program modules

- ☐ Determine whether you want all users to have access to OpenEdge product scripts and program modules. Proceed as follows:
 - Type **Y** to place OpenEdge scripts in `/usr/bin` **and** the destination pathname you defined in the “[Installation and working directories](#)” section on page B-3.
 - Type **N** to indicate that you want the Installation Utility to place the OpenEdge scripts only in the destination pathname you defined in the “[Installation and working directories](#)” section on page B-3.

Identical file exists in the installation directory

- ☐ Identify what you want the Installation Utility to do if an identical file is found to already exist in the installation directory:
 - Type **Y** to delete the file.
 - Type **N** to retain the file.

Command and Utility Reference

This appendix provides a reference to a select number of commands and utilities that are useful when performing tasks or understanding information presented earlier in this document.

The categories these command and utilities refer to are:

- [Administering and configuring Unified Broker products](#)
- [Installing and managing keys and digital certificates](#)

Administering and configuring Unified Broker products

This section highlights the following utilities and command that you can use to manage various Unified Broker products and their properties:

- [ASBMAN](#)
- [DBMAN](#)
- [Mergeprop](#)
- [NSMAN](#)
- [PROADSV](#)
- [WTBMAN](#)

ASBMAN

Starts, stops, adds AppServer agents, trims AppServer agents, and queries the status for an AppServer instance and its AppServer agent.

Operating system	Syntax
UNIX Windows	<pre>asbman { { -name AppServer-name { -kill -start -stop -query -addservers number-to-start -trimservers number-to-trim } [-host host-name -user user-name -user user-name] [-port port-number] } -help }</pre>

-name AppServer-name

This parameter is required. It specifies the name of an AppServer.

-kill

Stops and removes the AppServer from memory, no matter what it is doing.

-start

Starts an AppServer.

-stop

Tells the NameServer to stop itself.

Note: The AppServer stops only after completing any active client requests.

-query

Queries an AppServer for its status.

-addservers *number-to-start*

Specifies the number of additional servers to start.

-trimservers *number-to-trim*

Specifies the number of additional servers to trim.

-host *host-name*

Specifies the name of the machine where the AdminServer is running. If a host name is not specified, it defaults to the local host name.

-user *user-name*

Specifies a user name and prompts for a password. A user name and password are required only when you use the **-host** parameter and specify a remote host name. If you specify a remote host name with the **-host** parameter but do not specify a user name with the **-user** parameter, you receive a prompt for a username and password.

Windows supports three different formats for *user-name*:

- A user name as a simple text string, such as “mary”, implies a local user whose user account is defined on the local Windows server machine, which is the same machine that runs the AdminServer.

- A user name as an explicit local user name, in which the user account is defined on the same machine that runs the AdminServer, except the user name explicitly references the local machine domain, for example “.\mary”.
- A user name as a user account on a specific Windows domain. The general format is *Domain\User*, in which the *User* is a valid user account defined within the domain and the *Domain* is any valid Windows Server, including the one where the AdminServer is running.

`-port port-number`

Specifies the port number of the machine on which the AdminServer is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

DBMAN

Starts, stops, or queries a database. Before you can use the DBMAN command-line utility, you must use the Progress Explorer Database Configuration Tool to create the database configuration and store it in the `conmgr.properties` file.

Operating system	Syntax
UNIX Windows	<code>dbman [-host host-name -port port-number service-name -user user-name] -database db-name [-config config-name -start -stop -query]</code>

`-database db-name`

Specifies the name of the database you want to start. It must match the name of a database in the `conmgr.properties` file.

`-config config-name`

Specifies the name of the configuration with which you want to start the database.

-start

Starts the database *db-name* as defined by the configuration *config-name*.

-stop

Stops the database *db-name*.

-query

Queries the Connection Manager for the status of the database *db-name*.

-host *host-name*

Identifies the host machine where the AdminServer is running. The default is the local host. If your AdminServer is running on a remote host, you must use the **-host *host-name*** parameter to identify the host where the remote AdminServer is running.

-port *port-number*|*service-name*

Identifies the port that the AdminServer is listening on. If your AdminServer is running on a remote host, you must use the **-port *port-number*** parameter to identify the port on which the remote AdminServer is listening. The default port number is 20931.

-user *user-name*

If your AdminServer is running on a remote host, you must use the **-user *user-name*** parameter to supply a valid user name for that host. You will be prompted for the password.

For more information, see [OpenEdge Data Management: Database Administration](#).

Mergeprop

Provides a consistent means to manage and maintain the content of property files, by direct user action or programmatically. Property files store configuration information that identify and control the behavior of various components. The mergeprop program is located in the *OpenEdge-Install-Directory\bin*.

Presented through a command-line interface, the mergeprop utility is an alternative, fully supported tool by which you can update a property file when either the Progress Explorer tool is not available or you choose to use this approach.

The following table presents the mergeprop syntax.

Operating system	Syntax
UNIX Windows	<code>mergeprop -type <i>file_type</i></code> <code>[-action <i>operation_type</i> [<i>group_name</i>]]</code> <code>[-target <i>target_file</i>]</code> <code>[-delta <i>delta_file</i>]</code> <code>[-validate]</code> <code>[-nobackup]</code> <code>[-silent]</code> <code>[-recurse]</code>

Reference [Table C-1](#) for the details about valid values for argument variables.

All of the command switches identified in the previous syntax and presented in more detail in [Table C-1](#) can occur in any sequence following the mergeprop command.

Command switches and arguments

[Table C–1](#) summarizes the syntax elements used with the mergeprop command. Refer to the preceding section and the “[Mergeprop utility overview](#)” section on page 10–26 for additional descriptive information.

Table C–1: Command line input to the mergeprop command (1 of 2)

Switch	Arguments	Notes
-type (required)	ubroker database tools plugin none	Each argument (other than none) implies a specific target file in the properties directory. See the “File Type” section in Chapter 10 , “ Configuration ”.
-action ¹	update create delete list <i>group_name</i> listall <i>group_name</i>	If no action is specified, update is assumed by default. The list and listall actions require an additional argument, the name of the property group to be displayed (for example, ubroker.AS.asbroker1). Do not include the square brackets ([]) that enclose the group name in the ubroker.properties file. On update and create actions, groups listed with no properties in the delta file are ignored.
-target (optional)	Path to the property file to be modified.	If you are updating a property file that is in the <i>OpenEdge-Install-Dir/properties</i> subdirectory, you can omit this option. Only use this option when the property file you plan to update exists in a location other than the <i>OpenEdge-Install-Dir/properties</i> subdirectory.
-delta	Path to the delta file containing changes to be made.	File containing create, update, or delete changes.
-validate	None.	Performs all processing without actually making changes to the target file. This option lets you test for errors.

Table C–1: Command line input to the mergeprop command (2 of 2)

Switch	Arguments	Notes
-nobackup	None.	Does not create a backup to the target file before making changes. Unless you invoke this option, mergeprop saves a copy of the original target file in the same directory. The backup copy has a system-generated unique string appended to the name (for example, <code>ubroker.properties (31420040644533)</code>).
-silent	None.	Suppresses all messages.
-recurse	None.	Lists or deletes all groups, server groups, and configurations associated with the specified database.

1. Command switches can occur in any order following the mergeprop command.

NSMAN

Allows you to start and stop a NameServer and check the operational status of a NameServer that is located on either a local or remote NameServer instance. Unlike the Progress Explorer, the NSMAN utility does not support a means to view log files or delete configured NameServer instances.

Operating system	Syntax
UNIX Windows	<pre>{ { -name <i>name-server</i> { -kill -start -stop -query } [-host <i>host-name</i> -user <i>user-name</i> -user <i>user-name</i>] [-port <i>port-number</i>] } -help }</pre>

`-name name-server`

This parameter is required. It specifies the name of the NameServer.

`-kill`

Stops and removes the NameServer from memory, no matter what it is doing.

`-start`

Starts the NameServer.

`-stop`

Tells the NameServer to stop itself.

`-query`

Queries the NameServer for its status.

`-host host-name`

Specifies the name of the machine where the AdminServer is running. If a host name is not specified, it defaults to the local host name.

`-user user-name`

Specifies a user name and prompts for a password. A user name and password are required only when you use the `-host` parameter and specify a remote host name. If you specify a remote host name with the `-host` parameter but do not specify a user name with the `-user` parameter, you receive a prompt for a user-name and password.

Windows supports three different formats for *user-name*:

- A user name as a simple text string, such as “mary”, implies a local user whose user account is defined in the local Windows server machine, which is the same machine that runs the AdminServer.
- A user name as an explicit local user name, in which the user account is defined on the same machine that runs the AdminServer except the user name explicitly references the local machine domain, for example “\mary”.
- A user name as a user account on a specific Windows domain. The general format is *Domain\User*, in which the *User* is a valid user account defined within the domain and the *Domain* is any valid Windows Server, including the one where the AdminServer is running.

`-port port-number`

Specifies the port number of the machine on which the AdminServer is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

PROADSV

Supports various activities including starting up, shutting down, and querying the status of the current installation of an AdminServer.

Operating system	Syntax
UNIX	<pre>proadsv { { { -start { [-adminport <i>port-number</i>] } stop -query } [-port <i>port-number</i>] } -help }</pre>

`-start`

Starts the AdminServer.

`-admingroup groups`

Identifies a list of group names separated by a colon.

`-adminport port-number`

Specifies the port number used by the AdminServer for database broker communication. If a port number is not specified, the adminport defaults to port 7838.

`-f pluginsFile`

Points to an AdminServerPlugins.properties file by default. If a file is not defined for the -f, then this default is used.

`-propertyfile filename`

Database configuration information. The default value is *OpenEdge-Install-Dir/properties/conmgr.properties*.

`-requireusername`

Indicates that at least one user ID is required to be resolved for each AdminServer operation before each operation can be executed.

`-stop`

Stops the AdminServer.

`-query`

Displays the AdminServer status.

`-port port-number`

Specifies the listening port number. If a port number is not specified, the port defaults to 20931.

`-user username`

User who has been assigned AdminServer process privileges. The default is the current user.

`-password password`

The password associated with the *-user*.

`-help`

Displays the command-line help.

Table C–2 shows several options that you can use with `proadsv` to accomplish the corresponding tasks. Note that the examples use the port number 9999.

Table C–2: `proadsv` command-line options

AdminServer task	Commands	Examples
Start.	<code>-start</code>	<code>proadsv -start</code>
Specify the listening port.	<code>-port <i>port-number</i></code>	<code>proadsv -port 9999 -start</code>
Specify the database broker port.	<code>-adminport <i>port-number</i></code>	<code>proadsv -adminport 9998</code>
Stop.	<code>-stop</code>	<code>proadsv -stop</code>
Query.	<code>-query</code>	<code>proadsv -query</code>
Help.	<code>-help</code>	<code>proadsv -help</code>

Notes: The port numbers specified with the `-port` and `-adminport` options must be different.

If you are running multiple AdminServers, you must override both the default port and the default `adminport` settings.

WTBMAN

Controls the operation of a configured WebSpeed Transaction Server. The utility allows you to start a Transaction Server, query its status, start and stop additional WebSpeed Agents, trim by a certain number of agents, and shut down the Transaction Server. (WebSpeed only).

Operating system	Syntax
<p>UNIX Windows</p>	<pre>wtbman { { -name <i>transaction-server-name</i> { -kill -start -stop -query -addagents <i>number-to-start</i> -trimagents <i>number-to-trim</i> } [-host <i>host-name</i> -user <i>user-name</i> -user <i>user-name</i>] [-port <i>port-number</i>] } -help }</pre>

-name transaction-server-name

Specifies the name of a Transaction Server.

-kill

Stops and removes the Transaction Server from memory, no matter what it is doing.

-start

Starts the Transaction Server.

-stop

Stops the Transaction Server.

-query

Queries the Transaction Server for its status.

-addagents number-to-start

Specifies the number of additional agents to start.

`-trimagents number-to-trim`

Specifies the number of additional agents to trim.

`-host host-name`

Specifies the name of the machine where the AdminServer is running. If a host name is not specified, it defaults to the local host name.

`-user user-name`

Specifies a user name and prompts for a password when logging in to a remote machine. A user name and password are required only when you use the `-host` parameter and specify a remote host name. If you specify a remote host name with the `-host` parameter but do not specify a user name with the `-user` parameter, you receive a prompt for a user name and password.

`-port port-number`

Specifies the port number of the machine on which the AdminServer controlling the WebSpeed Transaction Server is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

Installing and managing keys and digital certificates

This section identifies each OpenEdge utility that allows you to install and manage keys and digital certificates (in key stores and certificate stores) so the components can use them. For Open Clients and clients of Progress Web services, OpenEdge provides utilities for some clients or relies on utilities provided by the client platform to manage the required certificate stores.

The utilities presented in this section are:

- `certutil`
- `genpassword`
- `mkehashfile`
- `pkiutil`

certutil

Provides all the functions necessary to install and manage root certificates from any Certification Authority (CA) as entries in the root certificate store of an OpenEdge client machine (located in *OpenEdge-Install-Dir\certs*).

Operating system	Syntax
UNIX Windows	<pre>certutil [-brief -verbose] { [-format { DER PEM }] -display cert-file [-format { DER PEM }] -import cert-file -list [alias ...] -remove alias ... }</pre>

-brief

Provides less information or as specified for the function.

-verbose

Provides more information or as specified for the function.

`-format { DER | PEM }`

Specifies the certificate format for the `-import` and `-display` functions. The default input format for a certificate is Privacy Enhanced Mail (PEM). Because some CAs issue root digital certificates in a binary format (DER), you must specify `-format DER` to import these certificates.

`-display cert-file`

Displays the digital certificate file information contained in the operating system disk file, *cert-file*. You must specify *cert-file* as a fully qualified operating system file path name. The `-verbose` option displays complete certificate information, and the `-brief` option displays less certificate information for each certificate store entry.

`-import cert-file`

Imports a trusted CA digital certificate from the disk file, *cert-file*. The *cert-file* argument must specify a fully qualified operating system file pathname. The function creates a root certificate store entry with a generated alias name and displays that alias name for view (specified by *alias* for other functions of this command).

Note: All root digital certificate store entry alias names are exactly eight hexadecimal characters in length and have a .0 (dot-zero) file extension. All other files in the root certificate store are ignored.

`-list [alias . . .]`

Displays a list of certificate store entries identified by each alias name (*alias*). You can specify multiple aliases, but you cannot use wild cards. If you specify no alias, `certutil` displays all entries in the certificate store. The `-verbose` option displays complete certificate information and the `-brief` option displays less certificate information per key store entry.

`-remove alias . . .`

Removes one or more certificate store entries that you specify by their *alias*. You cannot use wild cards. Moves each specified certificate store entry into the backup subdirectory and overwrites any previous backup subdirectory entry with the same alias name.

For more information on managing root certificates in the OpenEdge root certificate store, see [Chapter 9, “Managing OpenEdge Key and Certificate Stores.”](#)

genpassword

Accepts the clear-text value of a password and generates the encoded and encrypted form for the specified password.

Operating system	Syntax
UNIX Windows	<code>genpassword -password <i>text</i> [-verify <i>encrypted-password</i>]</code>

`-password text`

Where *text* is the clear-text value of the real password. When you specify the `-password` option alone, the tool displays a string of characters that represent the encrypted password value. You can then use this value directly to manually specify a required password as a property or command-line parameter value when manually configuring an OpenEdge client or server.

`-verify encrypted-password`

Where *encrypted-password* is the value of the encrypted password. When you specify the `-verify` option, the tool displays a message indicating if the real password and the encrypted password value match one another.

mkhashfile

Provides a simple way to install a root certificate in the OpenEdge root certificate store of a client machine. Such a certificate can be authorized by your own internal-use Certification Authority (CA) or by any CA that can provide you with a PEM-encoded certificate.

Operating system	Syntax
UNIX Windows	<code>mkhashfile <i>PEM-certificate-pathname</i></code>

PEM-certificate-pathname

Pathname of a PEM-encoded certificate file (typically with a .pem extension) containing a root certificate that you want to store in an OpenEdge root certificate store. The command creates a copy of this file with a hashed filename and places it in the *OpenEdge-install-dir/certs* directory of the client machine. The generated filename becomes the alias for the root certificate store entry.

You can use `certutil` to manage the root certificates that you install with this utility. For more information on managing root certificates in the OpenEdge root certificate store, see the “`certutil`” section on page C-15 and [Chapter 9, “Managing OpenEdge Key and Certificate Stores.”](#)

pkiutil

Provides all of the functions necessary to create and manage key store entries for OpenEdge SSL servers. It creates these entries from pairs of private keys and digital certificates that it stores in the OpenEdge server key store (located in *OpenEdge-Install-Dir\keys*).

Note: You must submit a public-key certificate request that is generated for each new key store entry that you want to create with this utility to a Certification Authority (CA). The CA then returns the necessary server (public-key) certificate for you to import and complete creation of the new key store entry.

Operating system	Syntax
UNIX Windows	<pre>pkiutil [-brief -verbose] { [-format { DER PEM }] -display cert-file [-format { DER PEM }] -import alias cert-file -list [alias ...] [-keysize size] -newreq alias -print alias -remove alias ... }</pre>

`-brief`

Provides less information or as specified for the function.

`-verbose`

Provides more information or as specified for the function.

`-format { DER | PEM }`

Specifies the certificate format for the `-import` and `-display` functions. The default input format for a certificate is Privacy Enhanced Mail (PEM). Because some CAs issue public-key certificates in a binary format (DER) you must specify `-format DER` to import these certificates.

`-display cert-file`

Displays the digital certificate file information contained in the operating system disk file, *cert-file*. You must specify *cert-file* as a fully qualified operating system file pathname. The `-verbose` option displays complete certificate information, and the `-brief` option displays less certificate information for each key store entry.

`-import alias cert-file`

Imports a CA-issued SSL server digital (public-key) certificate from the disk file, *cert-file*, pairs it with the `-newreq`-generated private key identified by the specified alias name (*alias*), and places the pair in the key store as a new entry identified by *alias*. The function prompts for the same password used to generate the public-key certificate request for this entry.

`-list [alias ...]`

Displays a list of key store entries identified by each alias name (*alias*). You can specify multiple aliases, but you cannot use wild cards. If you specify no alias, `pkutil` displays all entries in the key store. The `-verbose` option displays complete certificate information, and the `-brief` option displays less certificate information per key store entry.

`[-keysize size] -newreq alias`

Generates a new private/public-key pair and a corresponding public-key certificate request (suitable for submission to a CA), stored under the alias name specified by *alias*, and placed in the *OpenEdge-Install-Dir*\keys\requests directory.

You must specify an *alias* name between 1 and 39 characters long and use only the following characters:

- “0” to “9”
- “a” to “z”
- “A” to “Z”
- “_” and “-”

Note: The character “-” cannot be used as the first character.

The function prompts for a password with a minimum of four characters using any printable ASCII character. You must use this same password to later create and allow access to the key store entry generated from this certificate request.

When `pkiutil` generates the keys and certificate request for this function, by default it generates keys using the RSA asymmetric encryption algorithm with a 1024-bit key size. If you require a different key size, you can specify the number of bits to generate using the `-keysize` option (valid key sizes must be 512, 1024, or 2048 bits).

`-print alias`

Displays the public-key certificate request identified by *alias*.

`-remove alias . . .`

Removes one or more entries from the key store that you specify by their *alias*. You cannot use wild cards. Moves each specified key store entry into the backup subdirectory of the key store and overwrites any key store entry previously stored in the backup subdirectory with the same *alias*.

OpenEdge National Language Support

This appendix provides information about Progress messages in different languages, as described in the following sections:

- [Packaging](#)
- [Directory structure](#)
- [Contents of each directory](#)
- [Implementing regional support](#)
- [International databases](#)
- [Progress messages](#)
- [Environment variables of the SQL client](#)
- [Regional parameter files](#)
- [Progress.ini file and the Windows registry](#)

Packaging

OpenEdge users can select the language in which error and informational messages display when they are working in OpenEdge software. The Progress message file, PROMSGS, is translated into several different languages.

Some languages are shipped with the OpenEdge software and are selectable from the Language Choice dialog box during the installation. Additional languages in which PROMSGS can display are available to download from the Progress Download Center Web site available at <http://www.progress.com/esd>.

Table D–1 identifies the PROMSGS translations shipped to all OpenEdge users by default.

Table D–1: PROMSGS translations shipped with OpenEdge

Supported languages shipped with OpenEdge 10.1B software	
Czech (CZE)	Polish (POL)
Dutch (DUT)	Portuguese (POR)
English-American (AME)	Portuguese-Brazilian (BRZ)
English-International (ENG)	Spanish (SPA)
French (FRE)	Spanish-Latin American (SPL)
German (GER)	Swedish (SWE)
Italian (ITA)	—

Table D–2 identifies the additional languages in which PROMSGS is translated. These languages are available to download from the Progress Download Center at <http://www.progress.com/esd>.

Note: The Web site requires a valid account that your company must establish with Progress Software Corporation to access this information.

Table D–2: Supplemental PROMSGS translations available for download

Supported supplemental languages available to download	
Arabic (ARB)	Korean (KOR)
Chinese-Simplified (SCH)	Lithuanian (LIT)
Chinese-Traditional (TCH)	Norwegian (NOR)
Croatian (HRV)	Persian
Danish (DAN)	Romanian (ROM)
Finnish (FIN)	Russian (RUS)
Greek (GRE)	Serbian (SRB)
Hebrew (HBR)	Slovak (SVK)
Hungarian (HUN)	Slovenian (SVN)
Icelandic (ISL)	Thai (TAI)
Japanese (JPN)	Turkish (TUR)

Directory structure

The *OpenEdge-install-dir\prolang\Readme* file lists the subdirectories of the \prolang directory by language. Also included is helpful information about code-page tables in *convmap.dat*.

Contents of each directory

The `prolang` directory contains a subdirectory for each national language that you have chosen to install. Each language subdirectory can contain several files. [Table D–3](#) highlights and briefly describes the more important file types contained in a language subdirectory.

Table D–3: National language file descriptions

Filename	Description
<code>empty.db</code>	An empty, language-specific OpenEdge directory containing databases of various block sizes. The database is initialized with an appropriate code page and collation for your language.
<code>promsgs.lang¹</code>	A translated OpenEdge-related run-time messages file.
<code>lang.pf¹</code>	<p>A file containing the parameters used to start up OpenEdge with the appropriate settings for your region.</p> <p>For example, <code>engus.pf</code> contains parameters associated with English-American (AME). Also, the <code>startup.pf</code> file, which contains conventions used when your OpenEdge installation is started up, is located in this language subdirectory.</p>
<code>lang.df¹</code>	<p>A data definition file that can be loaded into an empty OpenEdge database to create a language-specific database. A database created by loading this file is identical to the empty database provided in this directory. You can use this file to create sort ordering variations in the database.</p> <p>For example, <code>ame88591.df</code> identifies an English-American (AME) data definition file.</p> <p>In the Asian directories, the file is named <code>_tran.df</code>.</p>
<code>progress.ini</code>	<p>A file containing the parameters used to start up OpenEdge with the appropriate regional settings. For example, the Japanese <code>progress.ini</code> contains Japanese font specifications.</p> <p>The <code>progress.ini</code> file is only installed in the language subdirectory that is identified as the default, or primary, language during the installation process. For information about establishing a language choice, see the “Language Choice” help topic in the Windows or UNIX online help.</p>

1. The variable `lang` stands for the language-specific reference.

Implementing regional support

The installation utility requires you to install at least one language.

During the installation you must choose a default language. If you want to change the default language after installing OpenEdge, see [OpenEdge Development: Internationalizing Applications](#) for detailed instructions.

The **International Settings** dialog box of the installation program creates an OpenEdge Startup (startup.pf) file to accommodate international conventions such as Date format, Number format, Character set, Collation, and Case.

Once you select the default language, the Installation Program copies the contents of the DLC\prolang directory to *OpenEdge-install-dir*. This affects your empty.db, promsgs, and progress.ini files.

See [OpenEdge Development: Internationalizing Applications](#) for more information about the following files:

- empty.db in multiple block sizes
- startup.pf
- promsgs
- progress.ini

International databases

As part of the installation media, OpenEdge supplies empty databases that support the language and collation standards of over thirty languages. The databases are located in the *OpenEdge-install-dir\prolang* subdirectories. See [Table D-1](#) for the subdirectory name for your language. For example, the empty database that you might use to build a Russian application is *OpenEdge-install-dir\prolang\rus\empty.db*.

These empty databases provide a database labelled with the appropriate code page and collation table for a language. However, if you are developing applications for a language or region that is not represented in *OpenEdge-install-dir\prolang*, OpenEdge has a utility, PROUTIL, that you can use to set up a unique database. See [OpenEdge Development: Internationalizing Applications](#) for detailed instructions.

Progress messages

The text used in Progress messages is contained in the PROMSGS file. OpenEdge provides various language editions of the PROMSGS file in the *OpenEdge-install-dir\prolang* subdirectories that you select during installation.

Note: Throughout the OpenEdge documentation set, Progress messages are also referred to as OpenEdge messages.

Each file has an extension that identifies its language.

To run OpenEdge with a certain version of PROMSGS, set the PROMSGS environment variable to the appropriate file. For example:

```
PROMSGS=C:\Progress\OpenEdge\prolang\ger\promsgs.ger
```

After you set the PROMSGS variable in the *progress.ini* file, you must run *ini2reg* to update the registry.

By default, Progress displays messages from *OpenEdge-install-dir\promsgs*.

File protection

OpenEdge incorporates specific file-protection measures to accommodate files associated with OpenEdge add-on products, which are OpenEdge products released independently of a point or major OpenEdge product release. Add-on products provide functionality that enhances the OpenEdge software product set and ensures that you have the most recent PROMSGS files. All OpenEdge products use one centralized method to display Progress messages contained in the PROMSGS file. With each OpenEdge add-on product you install, an updated PROMSGS file is installed into the destination directory. Add-on installation processes ensure that if the add-on product contains a newer PROMSGS file than the associated release, the following activities occur:

- The add-on product's PROMSGS file is compared with the product's PROMSGS file to determine which of the two files is newer.
- The newer file is copied to the OpenEdge directory.

Details about the installation and update of PROMSGS files

During the OpenEdge installation process, you select the languages that can be used during the product's execution. It is possible to have several translated PROMSGS files installed into the OpenEdge destination path\prolang subdirectory due to this selection process. During the installation process, the PROMSGS files for the language identified as the default language are copied from the OpenEdge destination path\prolang subdirectory to the *OpenEdge-install-dir* directory.

The PROMSGS files contain the most up-to-date messages at the time the OpenEdge product is released. However, the PROMSGS files are constantly being updated. Consequently, add-on products and OpenEdge install service packs that are released after the product release date can contain even more recently updated PROMSGS files. As each OpenEdge add-on product is installed, the installation program checks to ensure that the newest copy of the PROMSGS file is being used by all products; all products use the centrally located copy of the PROMSGS file stored in the *OpenEdge-install-dir* directory.

Procedures to protect PROMSGS files from being overwritten

OpenEdge protects PROMSGS files and any associated files, and ensures that you always have the most recent PROMSGS files:

- A file protection mechanism is part of the installation program and prohibits overwriting any PROMSGS file that already exists. If a PROMSGS file exists in the local directory, it is the latest version; therefore, there is no need to perform any file changes.
- The OpenEdge Installation program supports a versioning scheme that adds date information to the header of the PROMSGS file. The install program uses this date information to help determine the latest version of a PROMSGS file.

Procedures to ensure PROMSGS files are synchronized

In OpenEdge, PROMSGS files are considered to be either in synchronization or out of synchronization. These terms reflect the status of the date stamp associated with a PROMSGS file when the date in the header of the PROMSGS files located in the add-on directory is compared with the date in the header of the PROMSGS files currently installed in the *OpenEdge-install-dir* directory.

In OpenEdge, the installation processes are designed to compare and evaluate the date stamp information. A PROMSGS file is considered synchronized if, at the conclusion of any product installation process, the OpenEdge installation contains the PROMSGS file with the most current, or latest, date stamp. A PROMSGS file is considered out of synchronization, and therefore invalid, when the date stamp associated with the PROMSGS file does not display the most current date.

Table D–4 identifies the general installation sequence that can occur at a customer site when OpenEdge products and add-on products are installed. It illustrates how the PROMSGS files are compared, evaluated, and updated to ensure that the PROMSGS files are always synchronized.

Table D–4: PROMSGS file synchronization process (1 of 2)

Install sequence	When . . .	Then . . .
1.	OpenEdge products are initially installed.	The PROMSGS files associated with the languages selected by the user during the install process are installed to the <i>OpenEdge-install-dir</i> directory.
2.	An OpenEdge add-on product is installed.	<p>Date stamp information in the header of the existing PROMSGS file in the <i>OpenEdge-install-dir</i> directory is compared with the date stamp information in the header of the add-on product's PROMSGS file.</p> <p>If the PROMSGS file's date is later than the add-on product's PROMSGS file's date, the file is already synchronized and no changes occur.</p> <p>If the PROMSGS file's date is earlier than the add-on product's PROMSGS file's date, the add-on PROMSGS file replaces the existing PROMSGS file.</p>

Table D–4: PROMSGS file synchronization process

(2 of 2)

Install sequence	When . . .	Then . . .
3.	OpenEdge products are re-installed to add a new product and a new PROMSGS file.	<p>Two comparisons and their associated activities occur:</p> <ul style="list-style-type: none"> • If the re-installation process finds that a PROMSGS file exists, the existing PROMSGS file is not overwritten. • If, during the re-installation process, a new language is added, the PROMSGS file associated with that new language is installed into the <i>OpenEdge-install-dir</i> directory.
4.	Another OpenEdge add-on product is installed.	<p>The date stamp information in the header of the existing PROMSGS file in the <i>OpenEdge-install-dir</i> directory is compared with the date stamp information in the header of the add-on product's PROMSGS file.</p> <p>If the PROMSGS file's date is later than the add-on product's PROMSGS file's date, the file is already synchronized and no changes occur.</p> <p>If the PROMSGS file's date is earlier than the add-on product's PROMSGS file's date, the add-on PROMSGS file replaces the existing PROMSGS file.</p>

[Table D–5](#) illustrates another example of how this process works, using more detailed data for you to review.

The first column of [Table D–5](#) elaborates on the installation sequence outlined earlier in this section. In Step 1, the user initially installs OpenEdge Studio with a PROMSGS file for American English. The file header date of this newly installed PROMSGS file is 04/14/2007. In Step 2, when the user installs an add-on product, the add-on product installation compares the header date of its American English PROMSGS file, 04/15/2007, with the header date of the existing American English PROMSGS file, 04/14/2007. Since the header date of the PROMSGS file associated with the add-on product is later than the existing PROMSGS file, the PROMSGS file is updated or synchronized.

This example helps to illustrate the criterion for updating PROMSGS files. Only PROMSGS files associated with languages that are currently installed in the OpenEdge will be updated by the add-on installation process.

In Step 3, when the user installs another OpenEdge product, the OpenEdge AppServer, and identifies the Spanish PROMSGS file, the PROMSGS file with the date of 04/14/2008 is installed. This latter part of the example illustrates how the PROMSGS files can become out of sync per the date information in the respective headers.

Table D–5: Example of PROMSGS files being out of sync

Installation step order	Install . . .	And the PROMSGS file is . . .	Which contains this header date . . .
1.	A product such as OpenEdge Studio.	Installed for American English.	04/14/2008
2.	An add-on product.	Updated for American English.	04/15/2008
3.	A product such as Application Server.	Installed for Spanish PROMSGS.	04/14/2008

As [Table D–5](#) indicates, the installation of previously non-existing Spanish PROMSGS file dated 04/14/2008 into the OpenEdge installation is now out of synchronization with the updated American English PROMSGS file dated 04/15/2008, which updated the original American English PROMSGS file.

When an additional OpenEdge installation is performed and the OpenEdge Installation program detects that a PROMSGS language has been installed that did not previously exist as illustrated by Step 3 in [Table D–5](#), the OpenEdge installation program displays a message. This message indicates the following information:

- The add-on product name that contains the latest PROMSGS.
- The destination path of the add-on product.

The OpenEdge installation message only displays this message when it detects that add-on products have been installed and it reads a new file called addons. The addons file is a text file defined as a Windows initialization (.ini) file. This file is created and/or updated in the OpenEdge destination directory by the add-on installation program. To resynchronize your PROMSGS file, you must reinstall your add-on product.

Environment variables of the SQL client

OpenEdge contains the environment variables `SQL_CLIENT_CHARSET` and `SQL_CLIENT_CHARSET_PROMSGS` for SQL clients. You can use these variables to internationalize your applications. These environment variables determine the code page the client uses to display the following:

- Database data from the server.
- PROMSGS from the server.

Notes: You should set `SQL_CLIENT_CHARSET` only if you want clients to use a code page that is different from the code page the client operating system uses.

You should set `SQL_CLIENT_CHARSET_PROMSGS` only if you want run-time messages to use a code page that is different from either the code page the client operating system uses, or the code page set by `SQL_CLIENT_CHARSET`.

If you do not set either of these environment variables, then the SQL client code page will correspond to the language of the client operating system.

Code page client uses to display data

To display database data from the server, the client uses the code page set by `SQL_CLIENT_CHARSET`, if you have set this environment variable on the client machine. Otherwise, the client uses the code page of the client's operating system.

If you want to specify an SQL client code page that is different from the client operating system, you can set the `SQL_CLIENT_CHARSET` environment variable to the name of a Progress code page. When you set this variable to a code page, the SQL server converts text data that is sent **from the server to the client** to the code page set by `SQL_CLIENT_CHARSET`. The server also uses this code page when it converts text data that is sent **from the client to the server** to the server code page.

Code page client uses to display PROMSGS from the server

To display PROMSGS from the server, the client uses the code page set by `SQL_CLIENT_CHARSET_PROMSGS`, if you have set this environment variable. Otherwise, the client uses the code page set by `SQL_CLIENT_CHARSET`, if you have set this environment variable. If you have set neither of these environment variables, then the client uses the code page of the client's operating system.

If you want run-time messages at the SQL client to use a different code page from either the client operating system or the code page set by `SQL_CLIENT_CHARSET`, you can set the `SQL_CLIENT_CHARSET_PROMSGS` environment variable. When you set this variable to a code page, the SQL server converts run-time messages that are sent **from the server to the client** to the code page set by `SQL_CLIENT_CHARSET_PROMSGS`.

Note: The `SQL_CLIENT_CHARSET_PROMSGS` environment variable applies to `SQLDUMP` and `SQLLOAD`, which are actually SQL applications.

Regional parameter files

A useful technique for controlling an OpenEdge client session or server is to use a parameter file (`.pf`) with a startup or connection command. OpenEdge provides parameter files that set up OpenEdge sessions appropriately for a wide range of countries. You can use `.pf` files to specify the correct code-page settings for international applications. The setup of the `install-path\startup.pf` file is based on the installation options that you select.

The international parameter files are located in the `OpenEdge-install-dir\prolang` subdirectories. Parameter files are region- or country-specific rather than language-specific because parameter files set options that can vary from country to country. The `OpenEdge-install-dir\prolang\ger` directory has parameter files for Austria, Germany, and Switzerland to account for the differences among these German-speaking countries.

You should use the parameter file to make sure that the application and database are using the appropriate international settings. Typically, a parameter file for a internationalized application sets the parameters listed in [Table D-6](#).

Table D-6: **Startup parameters for a deployed application** (1 of 2)

Parameter	Description
Internal Code Page (<code>-cpinternal</code>)	The code page that OpenEdge uses in memory.
Stream Code Page (<code>-cpstream</code>)	The code page for stream I/O.
Case Code Page (<code>-cpcase</code>)	A case table in the <code>convmap.cp</code> file to use for uppercase/lowercase rules. Case rules are used by the CAPS and LC functions and by the ! formatting character.

Table D–6: Startup parameters for a deployed application (2 of 2)

Parameter	Description
Collation Code Page (-cpco11)	A table in the convmap.cp file to use for collation rules.
Date Format (-d)	The format in which an application displays dates. Specify the format as a three-character string, comprised of the letters d, m, y, in the order that you display the date.
Language (-lng)	The initial value for the CURRENT-LANGUAGE function, which determines from which r-code segment OpenEdge reads character-string constants. Specify the language as a character string in quotes.
European Numeric Format (-E)	OpenEdge interprets and displays a comma as a decimal separator and a period as a thousands separator for numeric values.
Fractional Separator (-numdec)	Specifies the numeric value of the character that represents, in formatted text, a number's decimal point. The default decimal point is a period (.).
Thousands Separator (-numsep)	Specifies the numeric value of the character that represents, in formatted text, the thousands separator in numbers. The default thousands separator is a comma (,).

Note: You can also use parameter files with OpenEdge utilities, for example, PROSHUT and PROUTIL.

Progress.ini file and the Windows registry

The `progress.ini` file sets up the user interface environment for Progress applications running in Windows and is an important part of deploying a localized application. It controls parts of the environment that vary across locales, and it allows you to specify colors and fonts.

OpenEdge supports the use of the Windows registry in Windows 2000, Windows 2003, and Windows XP Professional, and searches the registry first for system configuration information. However, you can still use an initialization file to ensure that deployed applications are configured correctly and consistently at customer sites. The information from the `.ini` file can be added to the registry upon installation.

Be sure to create and edit the `progress.ini` file on a system configured like the target system on which you intend to run it. For example, Japanese font names might use Japanese characters. You should edit a `progress.ini` file for use in Japan on a system supporting Japanese.

If you edit the `progress.ini` file, run `ini2reg` to update the registry.

The sections of the `progress.ini` file that can typically affect a localized application are the `[Startup]`, `[WinChar Startup]`, and `[fonts]` sections.

[Startup] and [WinChar Startup]

The `[Startup]` and the `[WinChar Startup]` sections contain OpenEdge environment-variable settings. The `[Startup]` section includes the variables for GUI clients, and the `[WinChar Startup]` section includes the variables for character clients, WebSpeed Agents, and the AppServer.

Table D–7 lists the environment variables that a typical localized application might need.

Table D–7: Environment variables

Environment variable	progress.ini file section	Description
DefaultFont	[Startup]	The default display font.
DefaultFixedFont	[Startup]	The default-fixed display font.
PrinterFont PrinterFont1 PrinterFont2 PrinterFont3	–	The font that the printer uses for the OpenEdge OUTPUT TO PRINTER statement.
PROMSGS	[Startup] [WinChar Startup]	The PROMSGS file that an application should use. For example, for a OpenEdge application to access Swedish translations of Progress error messages, set PROMSGS to c:\Progress\OpenEdge\prolang\swe\promsgs.swe.

[fonts]

The [fonts] section of the progress.ini sets the fonts that an OpenEdge application running on that system uses. The default progress.ini file that OpenEdge supplies in the United States sets the following fonts:

```
font0=Courier New, size=8
font1=MS Sans Serif, size=8
font2=Courier New, size=8
font3=Courier New, size=8
font4=MS Sans Serif, size=8
font5=MS Sans Serif, size=10
font6=MS Sans Serif, size=8, bold
font7=MS Sans Serif, size=8
```

These font settings might not apply to all the locales where your application will run. Some of the *OpenEdge-install-dir*\prolang directories contain progress.ini files with font settings appropriate for that country.

NameServer and Name Server Load Balancing Details

This appendix presents detailed information about the NameServer and NameServer load balancing feature, as outlined in the following sections:

- [NameServer overview](#)
- [Understanding load balancing](#)
- [Understanding server-level and connection-level fault tolerance](#)

NameServer overview

A *NameServer* is a single process that mediates client connections for a set of Unified Brokers that have registered with it. Any number and type of Unified Broker instance can register with a single NameServer, and each Unified Broker instance can register with exactly one NameServer. The NameServer that a broker instance registers with is the broker's *controlling NameServer*.

Note: Keep in mind that the NameServer is not required. The use of this element will depend on your implementation.

When a Unified Broker instance starts up, it can register with its controlling NameServer by sending its location and other configuration information. The NameServer uses this information to help resolve client connection requests. Part of this registration information is the Application Service that the Unified Broker supports. An *Application Service* is a designation for the particular business function that a Unified Broker provides. For more information on Unified Brokers, see the “[Working with Unified Brokers](#)” section on page 10–9 and the “[Unified Broker and Name Server relationship](#)” section on page E–3.

A NameServer can provide the following services for a Unified Broker product:

- **Location Transparency** — A requesting client does not need to know the network location of a Unified Broker instance. When a client attempts to create a connection to a Unified Broker instance, it first requests the connection from a NameServer to a broker that provides a specified Application Service. The NameServer then locates and assigns a broker to complete the connection that provides the specified Application Service.
- **Server-level fault tolerance and load balancing** — If you have installed the load-balancing option, you can provide server-level fault tolerance, where the NameServer can select from several Unified Broker instances to satisfy a client request. This option also allows you to balance connection load among multiple Unified Broker instances that provide the same Application Service. The NameServer then assigns connections among several Unified Broker instances based on a weight factor that you configure for each instance. For more information on server-level fault tolerance and how load balancing affects Unified Broker operations, see [Appendix E, “NameServer and Name Server Load Balancing Details.”](#) And, note that any NSMAN command that specifies a username typically also prompts for a password. For complete information about the syntax and options of the NSMAN command-line utility, see the “[NSMAN](#)” section on page C–8.

- **Connection-level fault tolerance** — You can also make multiple NameServer instances available to help ensure that at least one NameServer is available even if another fails. In this type of configuration, one of several possible NameServers resolves the connection request. Thus, you can provide connection-level fault tolerance for requesting clients. For more information on connection-level fault tolerance, see [Appendix E, “NameServer and Name Server Load Balancing Details.”](#)

For more information on how your Unified Broker product uses NameServers, see your product documentation.

Unified Broker and Name Server relationship

This section highlights the role of the Name Server as it specifically affects a Unified Broker.

Application Services

The Application Service that a Unified Broker provides is identified by a list of one or more names that you can optionally specify during broker configuration. Each Application Service name you specify is an arbitrary designation for the business function that the Unified Broker instance provides.

The NameServer maintains a separate Application Service name space for each Unified Broker type. So, an AppServer, OpenEdge Adapter for SonicMQ, WebSpeed Transaction Server, and DataServer instance can each register the same Application Service name with the same controlling NameServer without conflict. However with the load-balancing option, if you have multiple Unified Broker instances of the same type register the same Application Service name with the same controlling NameServer, you must guarantee that each Unified Broker instance provides exactly the same functionality. For AppServers and WebSpeed Transaction Servers, this means providing the same application procedures and database resources for all instances. For DataServers, this means accessing the same database for all instances.

If, for example, you use the same Application Service name to identify functionality on several AppServers, each of which supports different remote procedures and database connections, multiple requests from the same client application are likely to provide inconsistent results.

The default service

You do not have to designate an explicit Application Service for a Unified Broker. Instead, you can specify that the broker supports the *default service*. The default service is a special Application Service designation that supports default client connection requests. Thus, any Unified Broker that supports the default service and is of the appropriate type (AppServer, OpenEdge Adapter for SonicMQ, WebSpeed, or DataServer) can satisfy a connection request from a client that does not specify an Application Service name as part of its connection request.

Configuring NameServer communications

Both clients and Unified Brokers use User Datagram Protocol (UDP) to communicate with NameServers. UDP is an Internet standard, combined network layer, transport layer, and session layer protocol that provides a mechanism for connectionless communications. The connectionless nature of this protocol affords built-in benefits, such as the ability to implement fault-tolerant NameServers for client connections. For information on connection-level fault tolerance, see [Appendix E, “NameServer and Name Server Load Balancing Details.”](#)

To establish a Unified Broker connection, a Unified Broker client must specify the location of the NameServer that provides the connection. To register with a NameServer, a Unified Broker must specify the location of the NameServer where it needs to register. To specify the NameServer location, both components must know the UDP port number (or service name) on which the NameServer is listening and the host address of the machine where it resides.

Specifying NameServer ports and hosts

ABL clients provide this information when they specify the `CONNECT()` method, and AppServer Open Clients provide it using an equivalent Open Client method. You must specify this information for NameServers and other Unified Broker components in the Progress Explorer or directly in the Unified Broker properties file (`ubroker.properties`). For more information, see [Chapter 10, “Configuration.”](#)

If the component uses a UDP service name rather than a port number, you must also ensure that the `services` file on the component host (or Network Information Services (NIS), if used) properly defines the UDP service name.

Editing the services file

The `services` file stores the service name, port number, and protocol for various services on the network. For each NameServer that the component accesses, the `services` file on the component host must specify a service name associated with the NameServer UDP port number (default, 5162). Thus, if the component connects using the service name `namesv`, you might enter the service name definition for the `services` file extract shown in [Figure E-1](#).

<i>Server name</i>	<i>Port number</i>		<i>Protocol</i>
namesv	5162	/udp	
db1sv	2501	/tcp	
db2sv	2502	/tcp	
db3sv	2503	/tcp	

Figure E-1: Sample Unified Broker client services file

Understanding load balancing

Load balancing is a feature that allows client connection requests to be distributed among multiple Unified Broker instances that support the same Application Service. Load balancing is a NameServer option that comes installed with some products (for example, the WebSpeed Enterprise Transaction Server) or that you must install as an option with others (for example, the AppServer). If you have load balancing with your product, the NameServer assigns client connections to the appropriate Unified Broker instances based on weight factors that you specify.

If the weight factor that you specify for each Unified Broker instance is appropriate in relation to the others, the effect is to assign more connections to broker instances with greater resources, and thus to balance connection load among all the instances. You can set the load-balancing weight factor for each Unified Broker instance in the Progress Explorer or by editing the `priorityWeight` property in the `ubroker.properties` file.

Percentage weight factors

Properly specified, weight factors give some sense of the amount of work that an individual Unified Broker instance can handle. For example, [Table E-1](#) shows the effect of weight factors specified for three Unified Broker instances registered to support the same Application Service.

Table E-1: Weight factors based on percentage

Unified Broker name	Weight factor	Percent of time selected
AS1	20	20
AS2	20	20
AS3	60	60

The selection algorithm used by the NameServer guarantees that AS1 and AS2 are each selected 20% of the time and AS3 is selected 60% of the time. Thus, if the sum of weight factors for all Unified Broker instances that support the same application adds up to 100, each weight factor specifies the exact percentage of time that the NameServer selects the given Unified Broker instance over time.

Arbitrary sum weight factors

You can specify arbitrary weight factors as any sum of values (not necessarily 100), but the weight of each is always proportional to the sum, as shown in [Table E-2](#).

Table E-2: Weight factors based on arbitrary sums

Unified Broker name	Weight factor	Percent of time selected
AS1	2	2/7
AS2	2	2/7
AS3	3	3/7

Fail-over weight factor

You can also specify a fail-over weight factor of zero (0) for a Unified Broker instance that you want to accept connection requests when the NameServer finds no other Unified Broker instance available in the pool.

Understanding server-level and connection-level fault tolerance

By default, a Unified Broker instance relies on a single controlling NameServer to resolve client connection requests and a single Unified Broker instance to provide services to the client. You can configure the controlling NameServer so that multiple NameServer instances are available to resolve any client connection request, thus providing connection-level fault tolerance. If your product supports load balancing, you can also configure a single NameServer to resolve each connection request using multiple Unified Broker instances that support the same Application Service, thus providing server-level fault tolerance.

Figure E-2 shows the relationship between these configuration options.

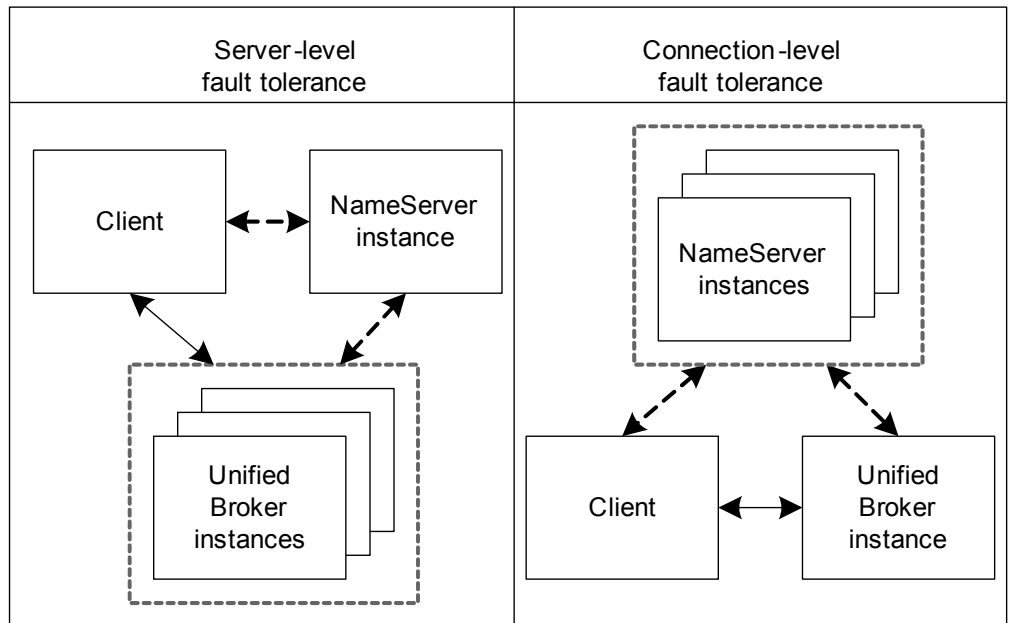


Figure E-2: Server-level and connection-level fault tolerance

These two levels of fault tolerance operate as follows:

- **Server-level fault tolerance** — Allows multiple Unified Broker instances to register with a NameServer for the same Application Service. A client requesting a connection is connected to one of several registered Unified Broker instances that the NameServer determines are available to provide the specified Application Service. If appropriate weight factors are specified, the NameServer also balances connection load among the several broker instances. For more information on load balancing, see the “[Understanding load balancing](#)” section on page E-5.
- **Connection-level fault tolerance** — Allows you to configure a collection of NameServers that work together to resolve a client connection request. You can use two different techniques, individually or together, to implement a fault-tolerant NameServer collection. This section describes these techniques and assumes that you are familiar with the documentation on using the NameServer with your Unified Broker product.

You can apply server-level and connection-level fault tolerance individually or together to achieve the level of fault tolerance that your application requires.

Connection-level fault tolerance

Connection-level fault tolerance enables a client (AppServer client, SonicMQ Adapter, WebSpeed Messenger, or DataServer client) to have its connection request satisfied by any NameServer from a set of related NameServers. You can configure NameServers for fault-tolerant operation using two different techniques, and you can use these techniques independently or together:

- **NameServer replication** — Where you configure multiple NameServer instances within a single subnet on different machines to listen on the same UDP port. Clients send connection requests and Unified Brokers send registration requests to all NameServer instances using UDP broadcasting. Using UDP broadcasting, the registration information from brokers is replicated on each NameServer that is listening (hence **NameServer replication**). Similarly, each client connection request is sent to each of the replicated NameServers.
- **NameServer neighbors** — Where you configure multiple NameServers on machines located in one or more subnets so that an initial NameServer instance receives the client connection request. If this initial NameServer cannot resolve the request, it passes the request on to a specified list of NameServer neighbors. These NameServer neighbors then attempt to resolve the connection request. Each NameServer neighbor represents the controlling NameServer for a separate Unified Broker instance.

Using either or both techniques, a client can receive multiple responses from multiple NameServers. The client uses the first response that indicates that the requested Application Service was found. A client only receives a connection error if all NameServers that respond indicate that the Application Service cannot be found.

In general, you can combine NameServer replication with NameServer neighbors to provide connection-level fault tolerance across an entire network. The following sections describe how to implement connection-level fault tolerance using these techniques.

Using UDP broadcasting

As described earlier, UDP is a connectionless protocol. This feature allows you to configure the following two types of communications with a NameServer:

- **Host request** — The client or Unified Broker sends a message directly to a NameServer residing on a specific host and listening on a specific port. The IP address represents the actual network location of a specific host. Only the NameServer on the specified host and listening on the specified port receives the message.
- **Broadcast request** — The client or Unified Broker sends a message specifying the UDP broadcast address of the NameServer host and the UDP port number on which the NameServer is listening. The UDP broadcast address represents the entire subnet where a host is located, and you can determine this address using the appropriate operating system commands from any host on the subnet. When a client or Unified Broker sends a UDP broadcast request, every NameServer on any host in the subnet that is listening on the specified port receives the message.

UDP broadcasting insulates the client and Unified Broker from having to know the exact host location of the NameServer. If there is some reason that you need to move the NameServer to a different machine in the same subnet, you can safely do it without having to change your client application or your Unified Broker configuration.

[Figure E-3](#) shows a client and Unified Broker using UDP broadcasting to communicate with the NameServer. Using the UDP broadcast address, 172.20.255.255, this client and Unified Broker can communicate with a NameServer running on any host in the 172.20 subnet.

Thus, you can use UDP broadcasting to support location transparency for a single NameServer. However, as [Figure E-3](#) implies, you can also use UDP broadcasting as the basis to support fault-tolerant NameServers using NameServer replication.

Using NameServer replication

UDP broadcasting supports NameServer replication by allowing a client or Unified Broker request to be received by multiple NameServers listening on the same UDP port and configured on different machines within the same subnet. Because every host on a subnet receives every broadcast request, one or more of these hosts can support a NameServer that receives and handles the same messages. This provides fault tolerance for both a client connection request and a Unified Broker registration request.



To configure and use replicated NameServers:

1. Run each NameServer instance on a separate host located within the same subnet.
2. Configure each NameServer instance to listen on the same UDP port.
3. Configure each client application to send its connection request and each Unified Broker to send its registration request using the subnet UDP broadcast address instead of the NameServer host address.

There is one broadcast address for each subnet. Using this address and the specified UDP port number, a client or Unified Broker sends a single request that is recognized by every NameServer listening on that port in the subnet.

Figure E–3 shows a client, a Unified Broker, and two replicated NameServers. The NameServer configurations shown for NameServer NS1 (above the dotted line) appear as they might in the ubroker.properties file for each host.

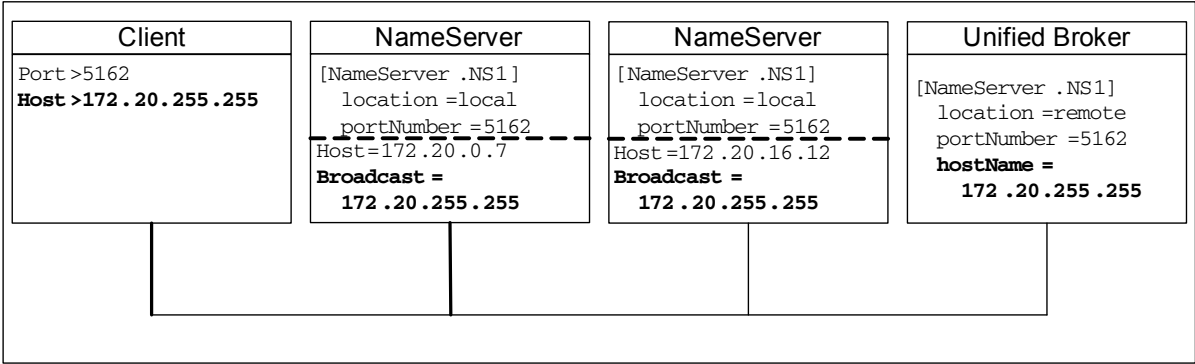


Figure E–3: NameServer replication

In [Figure E-3](#), one NameServer is located on a machine with the IP address 172.20.0.7 and another is located on a machine with the IP address 172.20.16.12. Both NameServers listen on UDP port 5162. The UDP broadcast address for these NameServers is 172.20.255.255. The Unified Broker is configured to register with a controlling NameServer remote from the Unified Broker machine using the UDP broadcast address 172.20.255.255 as the hostName. When the Unified Broker registers with its controlling NameServer using the UDP broadcast, it registers with both replicated NameServers. Similarly, when the client broadcasts its connection request using 172.20.255.255 as the NameServer host name, both replicated NameServers receive the request. The client uses the Unified Broker connection returned by the first NameServer that responds.

Note that if the NameServer at IP address 172.20.0.7 moves to a different host on the subnet, for example, with IP address 172.20.16.5, neither the client application nor the Unified Broker configuration has to change.



To configure and use NameServer replication:

1. Install the NameServer on each host within a single subnet where you want to replicate a NameServer configuration.
2. Configure each replicated NameServer to listen on the same UDP port number.
3. Determine the UDP broadcast address for the subnet where the NameServer hosts reside. For more information, see the [“Determining the broadcast address”](#) section on page E-12.
4. Configure each Unified Broker instance (AppServer, SonicMQ Adapter Broker, WebSpeed Transaction Server, or DataServer) to use a controlling NameServer as follows:
 - **Location** — Remote.
 - **Host name** — The UDP broadcast address that you determined from Step 3.
 - **Port number** — The UDP port number that you specified in Step 2.
5. Provide connection parameters to the client (AppServer, DataServer, or WebSpeed) that specify the required Application Service name, the broadcast address from Step 3, and the UDP port number that you specified in Step 2.

Determining the broadcast address

You can determine the broadcast address of a UNIX machine by using the netstat and ifconfig commands, as in the following example:

```
$ netstat -i
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 771334 0 771334 0 0 0
le0 1500 bali bali 15069970 286170 10019158 1 302211 0
$ ifconfig le0
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 172.20.0.7 netmask ffff0000 broadcast 172.20.255.255
```

This example shows that the IP address for bali is 172.20.0.7, and its broadcast address is 172.20.255.255.



To determine the broadcast address in Windows:

- 1. Enter the ipconfig command in the console, as shown in the following example:

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter CE2XPS1:

    IP Address. . . . . : 172.18.103.44
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : 172.18.0.19
```

- 2. For each bit in the Subnet Mask that has a value of 0, convert the corresponding bit in the IP Address to 1.

Note that the IP Address and Subnet Mask are composed of four dot-separated decimal numbers and each decimal number represents an 8-bit binary number. Also note that the decimal number 255 is 11111111 in binary.

In this example, the last two decimal digits of the Subnet Mask are zeros. Since the corresponding bits in the IP Address must be converted to 1, the last two decimal numbers of the IP Address should be 255. Therefore the broadcast address is 172.18.255.255. (For more information on determining broadcast addresses, consult with you network administrator.)

Using NameServer neighbors

In a typical environment where UDP broadcasting is used, there is at least one NameServer in each subnet where a Unified Broker exists. A client application that wants to make use of Unified Brokers in each subnet can make a separate connection request to the appropriate controlling NameServer for each Unified Broker. However, NameServer neighbors allow the client to make all of its connection requests using a single NameServer address.

NameServer neighbors are alternate NameServers that you specify as part of a NameServer configuration. When a NameServer receives a connection request from a client that it cannot resolve, it automatically passes the request to the specified NameServer neighbors to attempt the resolution.

NameServer neighbors support client applications in a similar way to replicated NameServers in that the client uses the first response returned by a NameServer, indicating that the requested Application Service was found. However, unlike replicated NameServers, each NameServer neighbor is typically the controlling NameServer for a separate and distinct Unified Broker configuration that might not support the same Application Services as the others.

You can configure each NameServer neighbor as a NameServer with its own set of NameServer neighbors. Thus, you can link NameServer neighbors to other NameServer neighbors for an arbitrary level of depth. You can also replicate both initial NameServers and NameServer neighbors for maximum fault tolerance.



To configure NameServer neighbors:

1. Define the NameServer neighbors as local NameServer instances on your network by using the Progress Explorer or by editing the `ubroker.properties` file. These neighbor NameServers can be defined in the same or different subnets in your network.
2. Specify their names as NameServer neighbors when you define the NameServer in the Progress Explorer, or assign the names in a comma-separated list as the value of the `neighborNameServers` property for the definition in the `ubroker.properties` file.

Figure E-4 shows how to configure NameServer neighbors and combines NameServer replication with NameServer neighbors.

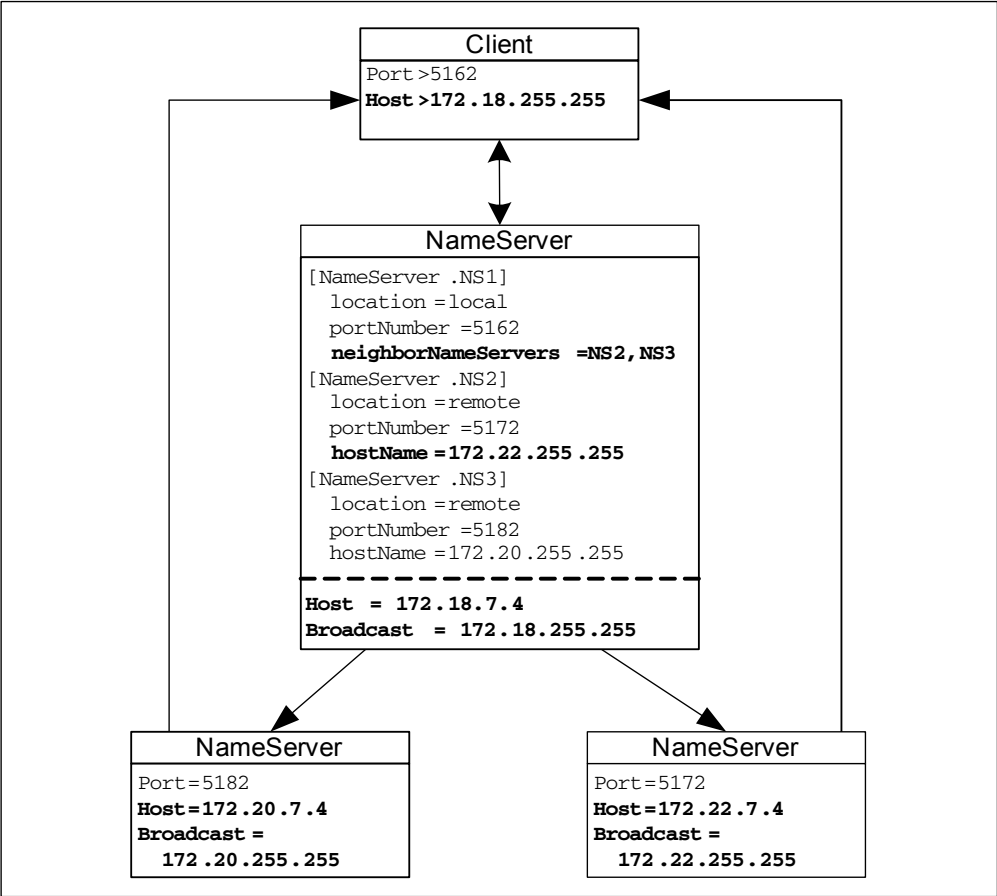


Figure E-4: NameServer neighbors

In [Figure E-4](#), if a client requests a connection to a Unified Broker that supports the Inventory Application Service using the 172.18.255.255 broadcast address, the NameServer at host 172.18.7.4 receives the request. If this NameServer has a Unified Broker that registered for the Inventory Application Service, it returns the location of that Unified Broker back to the client. If it does not have a Unified Broker that registered for the Inventory Application Service, this NameServer forwards the request to its neighbors, specified using the broadcast addresses 172.20.255.255 and 172.22.255.255. In this instance, the NameServers at hosts 172.20.7.4 and 172.22.7.4 receive the request.

Note: If you replicate these NameServers, all of the replicated NameServers in each subnet receive the request.

Neither of these NameServers has neighbors of their own, so both of them send a response back to the client. It does not matter if one of the NameServers does not know about the requested Application Service. The client uses the first positive acknowledgement from a NameServer and disregards the rest. The client application only receives an indication that the Application Service was not found if all responding NameServers indicate that the Application Service was not found.

Note that while NameServer neighbors provide the most benefit when using UDP broadcast, there is no requirement to do so. The `hostName` properties for NS2 and NS3 in [Figure E-4](#) can explicitly specify 172.20.7.4 and 172.22.7.4. You might want to use NameServer neighbors without broadcasting when you must tie together preconfigured NameServers, but where the performance implications of broadcasting outweigh the benefits. For more information, see the [“Performance implications of broadcasting”](#) section on page E-15.

Performance implications of broadcasting

When you use UDP to send a specific host request, only the specified host examines the message to determine what port it was sent to and whether an application (like the NameServer) is running and listening to that port on the host. However, when you use UDP broadcasting, either for NameServer replication or only to provide location transparency for a single NameServer, every host in the specified subnet examines the message for this same information.

Thus, using UDP broadcast might have a significant impact on the performance of your network if you have a large number of client applications that frequently connect to Unified Brokers. In deciding whether to use UDP broadcasting, you must weigh the benefits of location transparency for a single NameServer or replication of multiple NameServers against the impact on your Unified Broker and network performance.

Configuring OpenEdge NameServer instances

You can use the Progress Explorer to configure NameServer and Unified Broker instances, locally in Windows or remotely for both Windows and UNIX hosts. If you plan to configure instances directly on a UNIX host, you must edit the properties file (`ubroker.properties`) for each NameServer and Unified Broker instance directly on the host.

Note: The properties file that comes installed with your Unified Broker product includes one sample NameServer and Unified Broker instance for each type of Unified Broker that you can use as a guide. This section addresses using the NameServer and the Unified Broker instance. However, keep in mind that using the optional NameServer will depend on your company's implementation.

Downloading NameServer executables

If you need a NameServer executable, for example, for a different deployment platform, you can download it from the Progress Download Center available at www.progress.com/esd. Follow the instructions to download the OpenEdge NameServer executable for your OpenEdge release and platform.

Note: The Progress Download Center is located at <http://www.progress.com/esd>. You must have a valid user name and password to download products from this site. Contact a Progress Customer Service Representative to set up your Download Center account.

Order of configuration

To configure a NameServer and Unified Broker instance, you generally configure components in the following order:

1. Controlling NameServer and any replicated or neighbor NameServers.
2. Unified Broker product components.

In the Progress Explorer, you must have an initial configuration for the controlling NameServer instance to identify it when you configure your Unified Broker product instance. Editing the properties file, you can configure these components in any order. Whatever order you configure these components, you must have the controlling NameServer configured and running before clients can access your Unified Broker instance.

Configuring and using NameServer instances

You can configure two types of NameServer instances, determined by their functions, as the controlling NameServer for a Unified Broker instance:

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

When you configure a local NameServer instance, you can set all properties for the NameServer. When you configure a remote NameServer instance, you can only set its location (host and port) properties to identify the local NameServer instance that it references. When you want to start, stop, or obtain status on a running NameServer, you must always perform these actions on a local instance. You cannot start, stop, or obtain status on a remote NameServer instance.

How Unified Brokers use NameServer instances

To use a local NameServer instance as its controlling NameServer, a Unified Broker instance must run on the same machine where the local NameServer instance runs. Remote NameServer instances provide a way of having multiple Unified Broker instances use a controlling NameServer that runs on a different machine from the Unified Broker instances.

Whether local or remote, the NameServer instance that you define as the controlling NameServer must be defined on the same machine as the Unified Broker instance it controls. If the controlling NameServer instance is local, it runs on the same machine as the Unified Broker. If the controlling NameServer instance is remote, it references a NameServer running locally on a machine that is remote from the Unified Broker.

Thus, any remote NameServer instance you define must have a corresponding local NameServer instance defined on the machine where it runs, and you must define one such remote NameServer instance on each remote machine where a Unified Broker instance references this same corresponding local NameServer instance as its controlling NameServer.

NameServer instances and client connections

Unified Broker clients do not use local and remote NameServer instances. Clients must direct all connection requests to a NameServer on the machine where it runs, that is, to a NameServer where it is defined as a local instance.

Configuring the NameServer in the Progress Explorer

You can use the Progress Explorer to define and configure a NameServer instance. (See the Progress Explorer online help for detailed information.) When you configure a NameServer instance, you can do the following things in each of the property categories:

- **Location** — Sets port numbers and connection types.
- **General** — Sets the working directory, whether the NameServer starts automatically, and when the NameServer unregisters brokers.
- **Logging setting** — Sets how the NameServer logs events.
- **Advanced features** — Specifies one or more NameServers from the already-configured NameServers to serve as Neighboring NameServers. These are the NameServers that provide connection-level fault tolerance for a Unified Broker application. For more information, see [Appendix E, “NameServer and Name Server Load Balancing Details.”](#)
- **Environment variables** — Sets environment variables for NameServer execution. Windows users refer to [Chapter 7, “Working in the OpenEdge Environment in Windows,”](#) and UNIX users refer to [Chapter 8, “Working in the OpenEdge Environment on UNIX”](#) for more information.

Starting and managing a NameServer using the Progress Explorer

When you start a NameServer instance with the Progress Explorer, a brief status displays for the selected NameServer in the right pane showing that the NameServer is running.

Using the Progress Explorer you can also invoke the following management functions for the running NameServer instance:

- Stop the NameServer.
- Check the operational status of the NameServer.
- View the log file for the NameServer.
- Delete the NameServer instance.

Note: Before you can delete a NameServer instance, you must stop the NameServer and have no running Unified Broker instance still reference the NameServer as its controlling NameServer.

For more information on invoking NameServer management functions in the Progress Explorer, see the Progress Explorer online help.

Configuration Models

This appendix provides information about different configuration models you can reference and details about running OpenEdge installations in a network environment, as described in the following sections:

- [Shared-memory configurations](#)
- [Client/server configurations](#)
- [Client/server and OpenEdge AppServer in the network environment](#)
- [Preparing to run OpenEdge in a TCP/IP network](#)

Shared-memory configurations

Shared memory is an area in system memory that multiple users can access concurrently. OpenEdge keeps resources shared by all database users in shared memory and lets multiple servers access those resources efficiently. Optionally, additional asynchronous I/O processes can off load I/O operations from each server, further improving resource utilization and performance.

Local clients running multi-user OpenEdge can access database resources directly, rather than through a database server. This eliminates client/server message exchange and task-switching overhead. Database requests do not have to be queued until a server can process them. Local direct-access clients are known as *self-service clients*.

To run OpenEdge over a network, you need information regarding network-related system files, network configuration, and the startup parameters required to start remote clients. For more information about the network files and configuration, see the [“Client/server and OpenEdge AppServer in the network environment”](#) section on page F-9 and the [“Preparing to run OpenEdge in a TCP/IP network”](#) section on page F-15. For information about starting remote clients, see [Chapter 11, “Starting and Running OpenEdge.”](#)

Shared-memory architecture

Figure F–1 shows the shared-memory OpenEdge architecture.

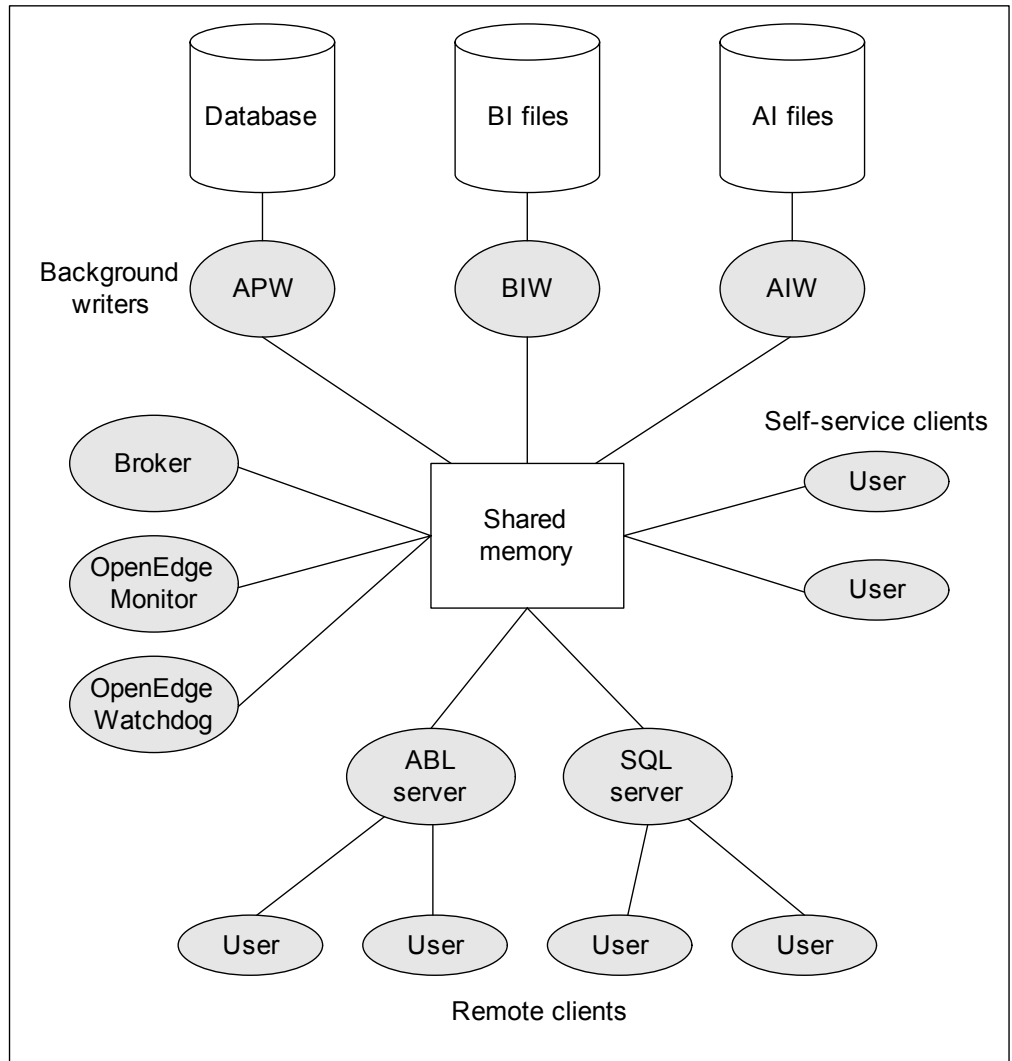


Figure F–1: Shared-memory OpenEdge architecture

The following sections explain the components of the architecture.

Broker

The initial database server process is identified as the *broker* (`_mprosrv`). The broker process manages shared resources and starts servers for remote users, as needed.

OpenEdge Database Monitor utility

The OpenEdge Database Monitor utility (`_dbagent`) displays performance and usage information about database status and activity.

For more information about the Database Monitor utility, see the description of the PROMON utility in *OpenEdge Data Management: Database Administration*.

OpenEdge Watchdog utility

If a process terminates improperly, it can maintain a lock on a record or shared-memory structure. This can impact database concurrency. The OpenEdge Watchdog utility detects processes that have terminated improperly and cleans up after them.

At regular intervals, the Watchdog utility checks for processes that have terminated unexpectedly. If it finds one, it releases any locks or shared-memory structures that the process might hold.

The Watchdog utility checks for inactive processes approximately once every 10 seconds. It also checks for self-service clients that are no longer active, releases all the appropriate record locks, backs out of any live transactions, and releases any shared-memory locks. If a server process terminates unexpectedly, the Watchdog utility disconnects and cleans up the server's remote clients.

For more information about the Watchdog utility, see the description and other details about the PROWDG utility in *OpenEdge Data Management: Database Administration*.

Background writers

The OpenEdge Enterprise RDBMS offers three background writer processes that improve performance. These processes continually perform certain housekeeping functions in the background. Because these functions are performed regularly by the dedicated background writer processes, client and server processes rarely have to wait for these functions to be performed.

The three types of background writers, asynchronous page writers, before-image writers, and after-image writers, are described in the “[Processes in Windows platforms](#)” section on page 6–13 and in the “[Processes on UNIX platforms](#)” section on page 6–18. The AdminService starts the background writers if the AdminService has been configured to do this by Progress Explorer. For more information about background writers, see *[OpenEdge Data Management: Database Administration](#)*.

Client/server configurations

Wherever it runs, multi-user OpenEdge functions in a client/server architecture. On a single machine, OpenEdge provides multi-user access to a database by using a separate client process for each user. In a client/server configuration, one or more clients access the database through a server. The server provides a connection to the database through the shared memory. While separate and distinct, the OpenEdge client and server processes compete for the same machine resources.

In client/server configurations, the client application and the database server are separate processes. Client processes can be local or remote.

The OpenEdge user interface and OpenEdge applications execute in the client session, sending requests to the OpenEdge server. The OpenEdge server accesses the database on behalf of each client session.

Terminology

This section introduces the terminology used to describe client/server configurations.

Application workstation

An *application workstation* is any node that runs one or more OpenEdge clients. For example, Windows 2000 is a multi-processor, so you can start more than one client on an application workstation. Depending on its operating system and configuration, an application workstation might run local clients and servers as well.

Database server machine

A *database server machine* is any node that runs one or more OpenEdge servers for local or remote OpenEdge clients.

Network file server

A *network file server* is any node that provides shared services including file, printing, and security services to other nodes, including application workstations and database server machines. A network file server usually provides these services by allowing other nodes to access its local files and printers as if they were local to the other nodes. For example, OpenEdge clients can run application procedures and OpenEdge servers can access database files stored on a remote network file server.

A network operating system (NOS) is a network environment that includes one or more network file servers that provide a common set of resource sharing and security services to other nodes. A network file server usually runs the *kernel* of an NOS, the program that controls access to shared network resources. Depending on its operating system, a network file server might also run one or more OpenEdge database clients and servers.

Although Progress Software Corporation recommends that you store the database on a disk locally attached to the database server machine, you can store the database on a network file server. Clients can access shared application code and communicate with the database server. Depending on your application and network environment, however, you might lose database integrity.

Note that OpenEdge often runs in local area networks (LANs) that have no network file servers. On these LANs, application workstations can access only locally stored procedures, and database server machines can access only locally stored databases. However, the application workstations and database server machines can communicate with each other as remote processes.

Single-process database server machine

A *single-process database server machine* is a node that runs only one server process for each database, providing access to that database for self-service clients only.

Multi-process database server machine

A *multi-process database server machine* is a node that runs multiple server processes for each database, providing multiple data paths to the database. Each server queues and runs requests for one or more clients. A separate broker process starts a new server for each additional client (or set of clients, in specified increments) that access the database. For more information on server machine configurations, see the “[Shared-memory configurations](#)” section on page F-2.

You can dedicate all the resources of a database server machine to run database servers. However, depending on your application and operating system, you can also run local clients and remote clients for other database server machines.

Simple client/server configurations

[Figure F-2](#) shows a simple client/server configuration, where the client and server components both run on a single system.

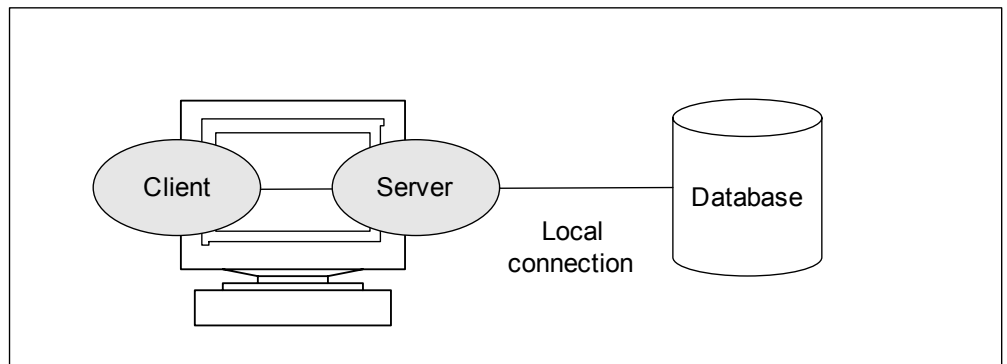


Figure F-2: Simple client/server configuration

Figure F-3 shows a multiple system client/server configuration. In this configuration, the server runs on the system where the database resides. The clients run on remote systems, accessing the database through the server system

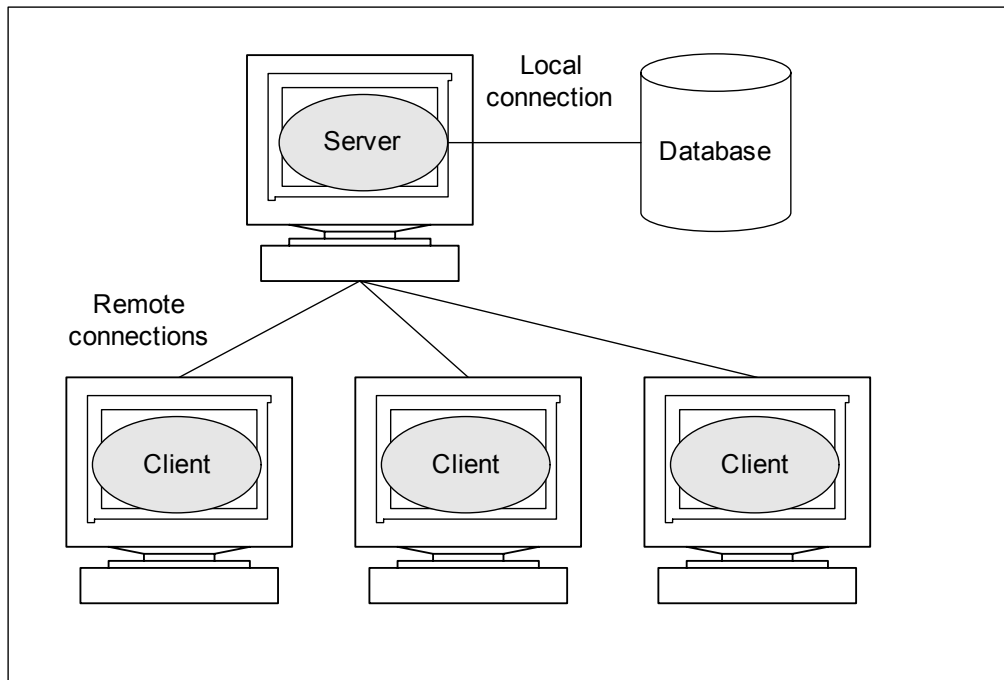


Figure F-3: Multiple system client/server configuration

Client/server and OpenEdge AppServer in the network environment

The OpenEdge client/server architecture fits naturally into a network environment, allowing clients and servers to run together in many different (heterogeneous) hardware and operating system environments. On a network, the OpenEdge client and server processes are distributed to separate nodes where they communicate through a common network protocol. Some nodes run client processes, while others run server processes. One advantage of this is that adding users or databases has minimal impact on the machine resources used by others. Each has its own resources devoted only to its client or server tasks. Another advantage is that a single OpenEdge application can take advantage of the strengths of a multi-machine, multi-operating system environment, without regard to differences in file resources on the separate machines. Remote OpenEdge clients and servers interact transparently, regardless of the type of machine environment in which they run. The result is a cooperative application environment with many more possibilities for expansion.

OpenEdge TCP network support

OpenEdge allows client and server operation among Windows systems that communicate using TCP/IP. In an OpenEdge AppServer configuration, the client connection to the application server is always TCP. The OpenEdge AppServer supports all of the client/server network types for the connection of the application server to the database server.

Figure F-4 shows the simplest OpenEdge network configuration—a database server machine and an application workstation. Although the figure shows only one database server machine and workstation, there can be more than one of each.

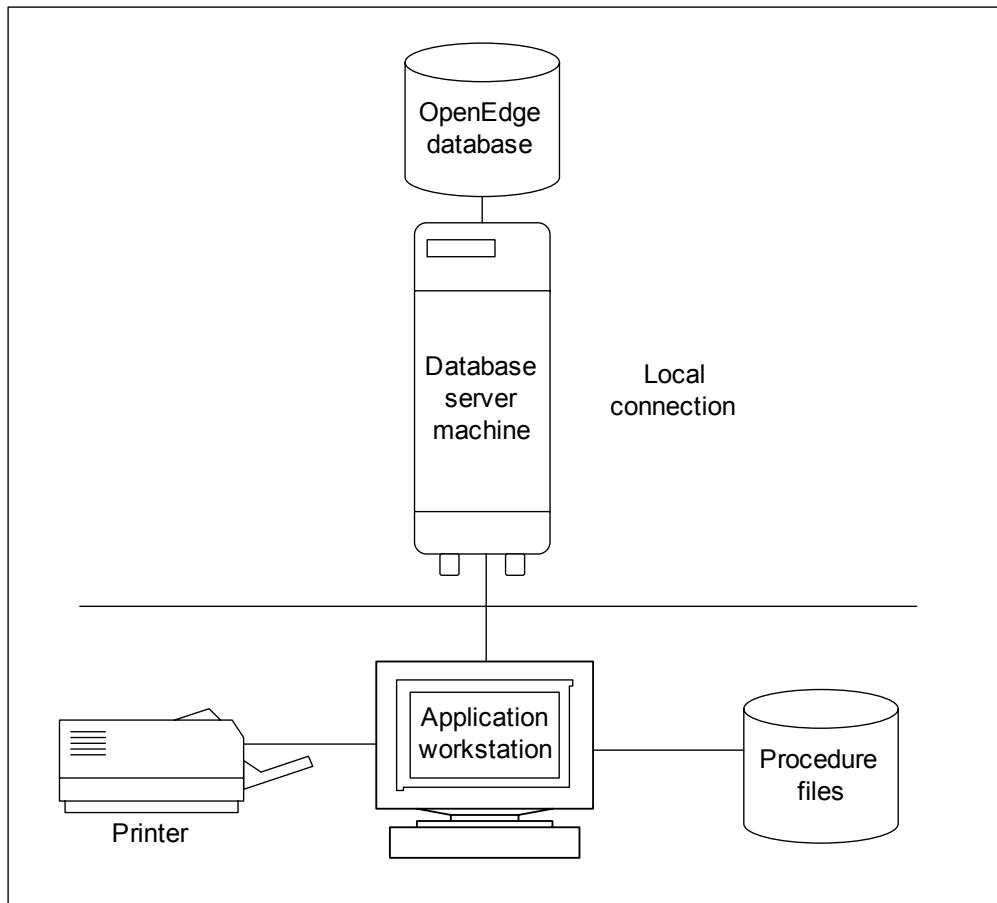


Figure F-4: Simple OpenEdge network configuration

This configuration is typical of TCP/IP networks without file servers. There are no shared resources except the OpenEdge database. The application workstation and database server machine each have a hard disk. A printer is also attached to the application workstation. OpenEdge is installed on each node.

A workstation in this configuration often supports multiple users and clients (for example, a system with multiple terminals) who share the local printer and OpenEdge application. The database server machine is usually a high-performance back-end processor that can also support local self-service clients. This network configuration, with the OpenEdge database local to the database server machine, ensures full database integrity. With all files stored local to each node, it generally (but not always) provides the highest performance on a LAN.

Figure F-5 shows a dedicated network file server, dedicated OpenEdge database server machine, and application workstations. Although the figure shows a limited number of workstations, file servers, and database server machines, there can be more of each.

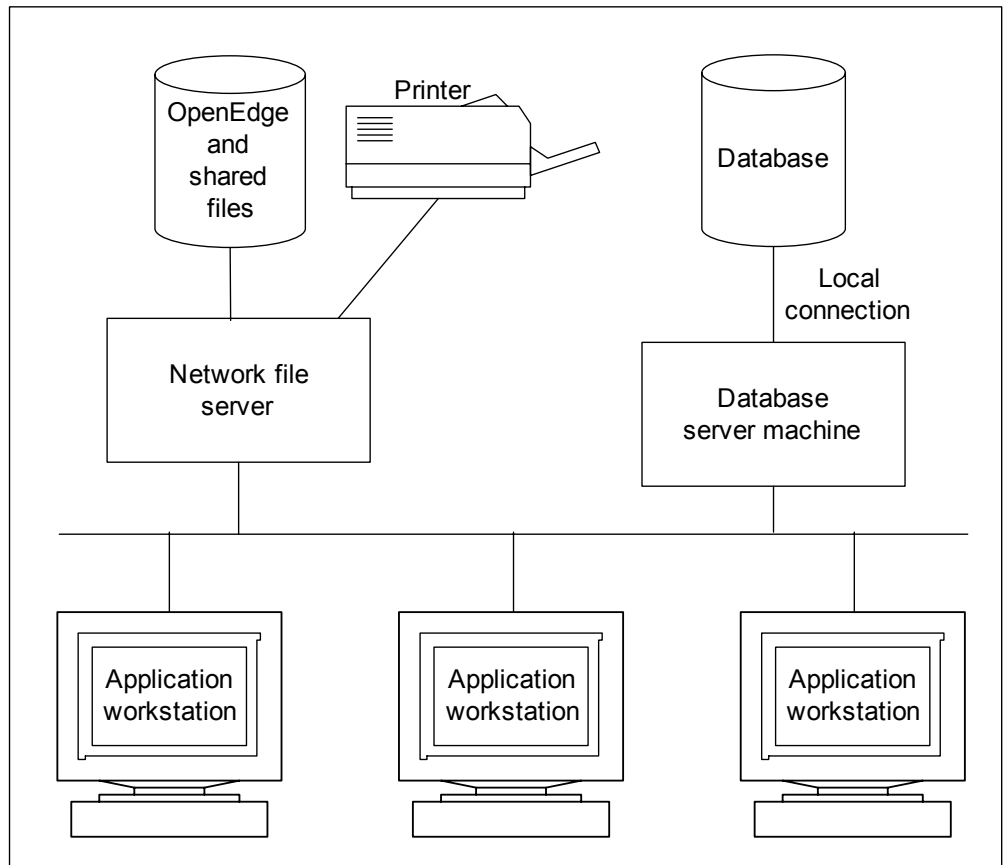


Figure F-5: Network file server for application files

This is a configuration typical of PC LANs with file servers and network operating systems. A hard disk and a printer are attached to the network file server, and an additional hard disk is attached to the OpenEdge database server machine. The OpenEdge database is on the disk drive that is locally attached to the OpenEdge database server machine. OpenEdge and all application procedures are installed on the file server and shared by all other nodes.

This network configuration ensures full database integrity and high performance, limited only by network and application performance capabilities.

Figure F-6 shows a network file server doubling as an OpenEdge database server machine and disk-optional application workstations. Although the figure shows a limited number of workstations, file servers, and database server machines, there can be more of each.

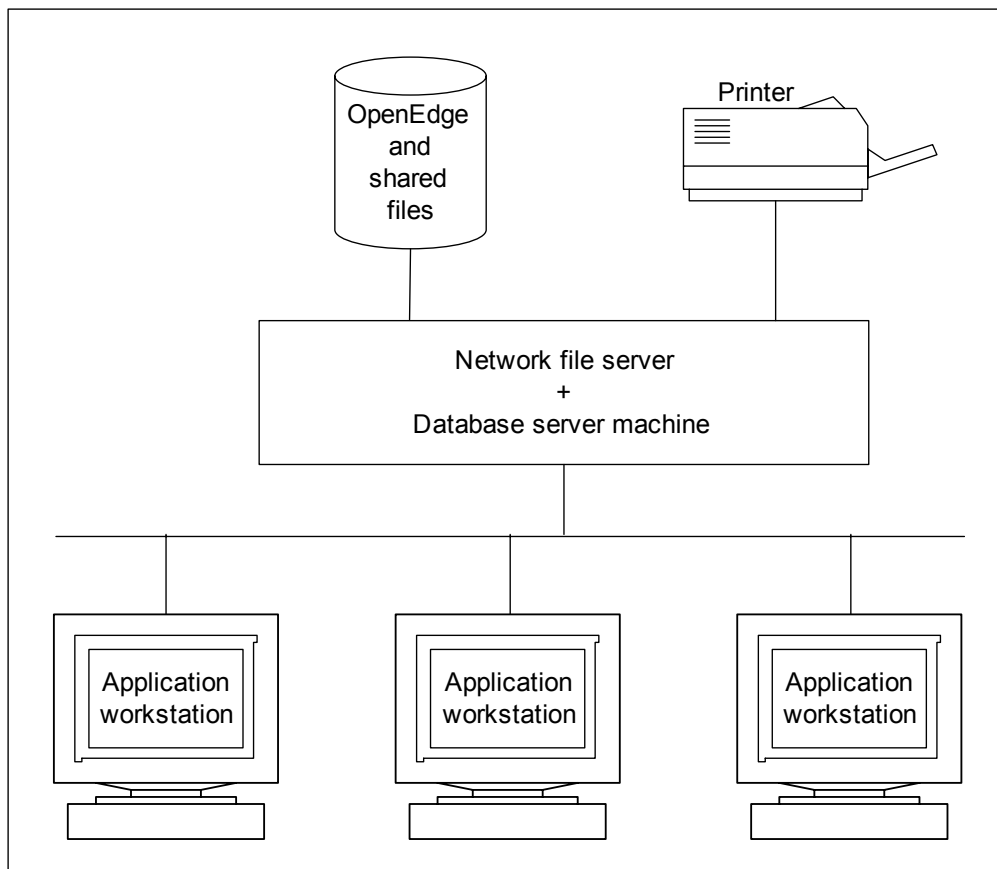


Figure F-6: Network file server as a database server

This is a configuration you might find on a PC LAN with a powerful file server running a multi-tasking operating system. OpenEdge, application procedures, and the OpenEdge database are all installed on the file server and are shared by the other nodes.

This network configuration provides full database integrity and acceptable performance on a file server with high-speed CPU and I/O resources.

Note: Avoid doubling a network file server as a database server machine on low-capacity nodes or on nodes where the database server machine can run only in an emulated environment.

Figure F-7 shows database files residing on a network file server.

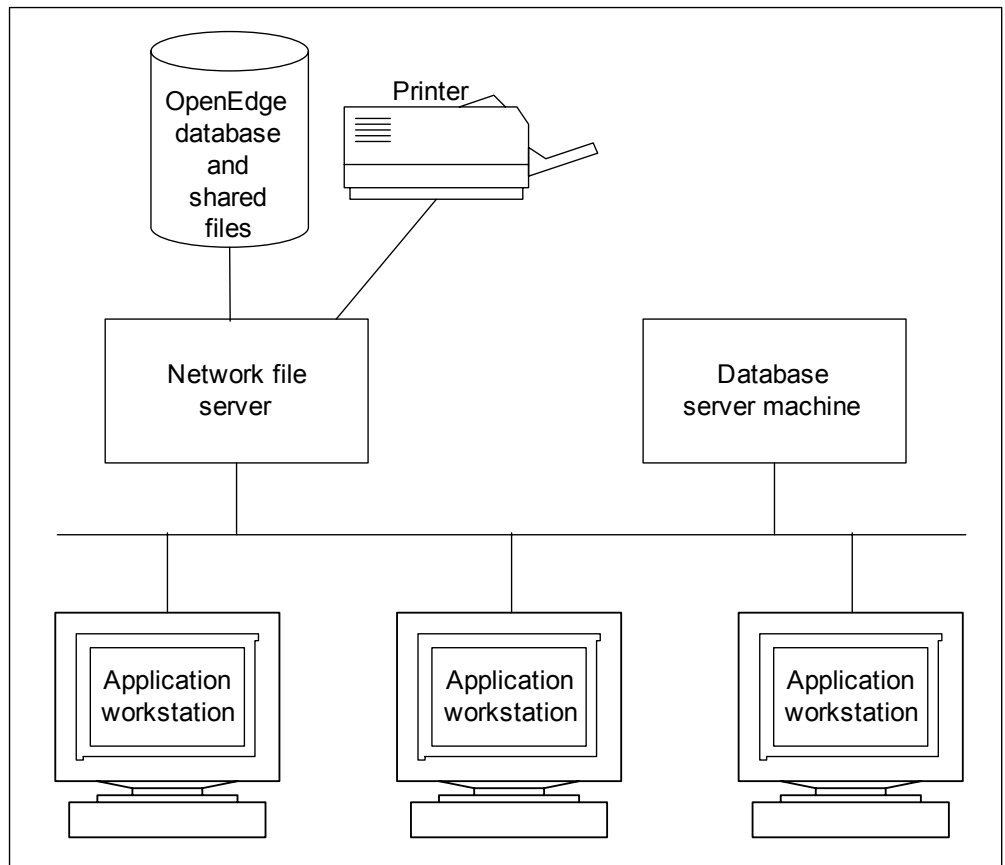


Figure F-7: Network file server for application and database files

This network configuration runs the risk of compromising database integrity if the network file server or database server machine crashes, because the before-image (BI) file is on the network file server, making synchronous writes to it impossible. Performance also depends on whether network file server I/O efficiency compensates for traffic across the network.

An application server running on the application server machine connects through shared memory to an OpenEdge database and has access to a set of procedure files. An ABL application runs at the application workstation, connects to the application server running on the OpenEdge AppServer machine, and sends the requests to the application server to run remote procedures. The procedure execution and database access occur in a remote OpenEdge session context.

Figure F-8 shows the simplest LAN configuration for OpenEdge on a network.

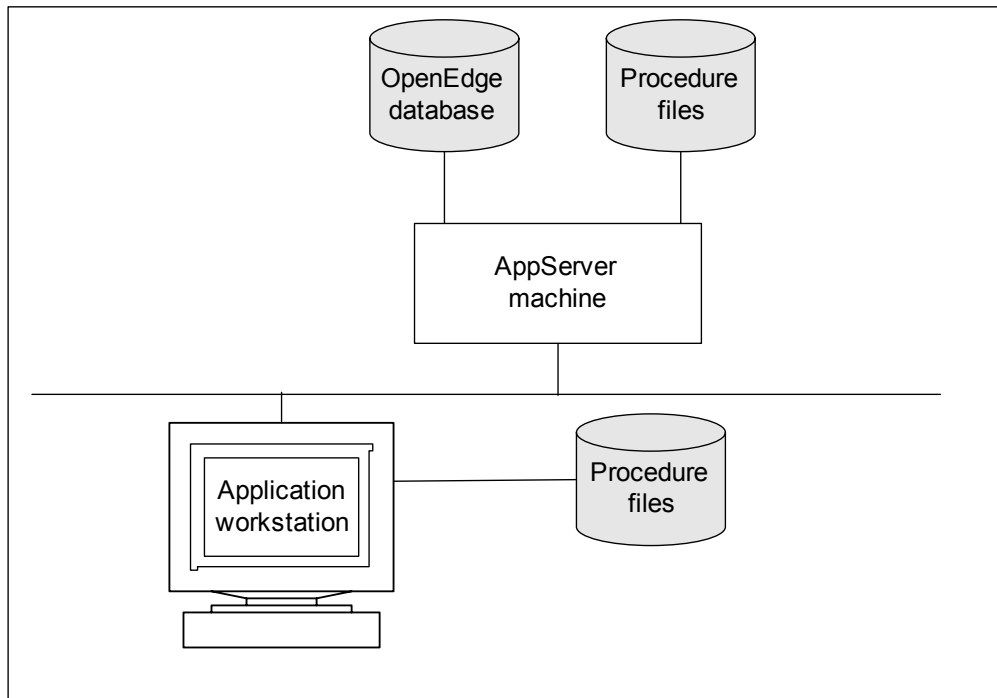


Figure F-8: LAN configuration with the OpenEdge AppServer

In more complex implementations of the OpenEdge AppServer, an application server can connect to another application server in order to connect with a database. For more information about the OpenEdge AppServer, see *OpenEdge Getting Started: Application and Integration Services*.

Preparing to run OpenEdge in a TCP/IP network

You can make OpenEdge operational in a network environment by following these guidelines:

- Identify and configure the nodes on your network for use as application workstations, database server machines, application server machines, and network file servers.
- Install OpenEdge on each node, or if your network has a network file server, install OpenEdge on the file server. For more information, see the [“Sharing an OpenEdge installation on a network overview”](#) section on page 4–50.
- If any application workstations and database server machines have incompatible processors or operating systems, you must install the appropriate OpenEdge product on each node.
- Set up network system files on each node.
- If you are using a network file server, make its resources, including printers and directories, available to all other nodes that require them.
- If you installed OpenEdge on a network file server, you might want to distribute the appropriate OpenEdge system files to the compatible application workstations and database server machines that use them. This takes advantage of networks where the local file and data access is faster than using the network.
- Set up your OpenEdge databases on each file server, database server, and application server machine.

Installing OpenEdge on your TCP/IP network

When installing OpenEdge on your network, keep these two basic considerations in mind:

- Where to place your database.
- Where to place your OpenEdge executables and r-code files.

Locating your database

Place your database on the hard disk of the machine that runs the OpenEdge server. If you place the database on a remote file server, synchronous writes are lost along with your database’s integrity, in the event of a system crash.

Synchronous writes ensure database integrity by flushing system buffers directly to disk. This is especially important for maintaining the before-image (BI) file. Therefore, if you must keep your database separate from the database server machine, use the Before-image Filename startup parameter to keep the before-image file local to the database server.

Note: Remote OpenEdge clients do not have to be concerned about synchronous writes because they do not write to the database.

Typical TCP/IP configuration with a hard disk on each machine

Figure F-9 shows the configuration for a typical network when there is a hard disk on each machine and no file server is used.

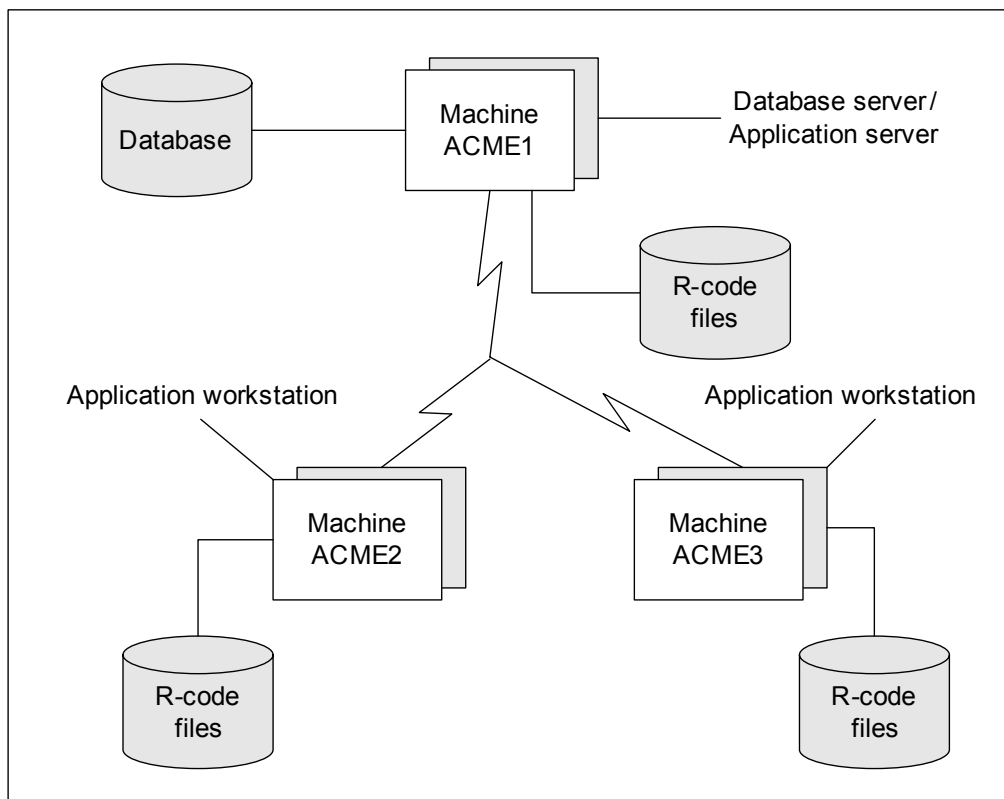


Figure F-9: Typical TCP/IP configuration (file server not used)

When you use this configuration, you must install OpenEdge on each machine in the network. In [Figure F-9](#), the client machines do not have to be running the same operating system.

Setting up network files to run OpenEdge

There are several files you must check, and modify if necessary, before you can run OpenEdge on your network. The filenames and locations might differ for different operating systems and TCP/IP implementations, but the functional contents are identical. [Table F-1](#) lists these files.

Table F-1: TCP/IP network files

File	Purpose
hosts	Lists machine names/network addresses.
services	Lists OpenEdge server/port number.
protocols	Defines system protocols.

Configuring OpenEdge on a network operating system

This section describes preparations that you can make to promote efficient and reliable OpenEdge operation in a network operating system (NOS) environment, that is, a network environment that includes one or more network file servers that provide a common set of resource sharing and security services to other nodes. This section describes some of the more general considerations.

Making network resources available

Once you have installed OpenEdge, you must make sure that each application workstation and application server machine has access to OpenEdge system files, application files, and any other necessary network resources (such as printers). Each NOS provides a set of commands or utilities to make these resources available across the network. In general, you set up pointers to remote resources so that each workstation can access them as though they were local to the workstation. These pointers can be in the form of logical drives in Windows nodes, or mounted directory paths on UNIX nodes.

For more information on making network resources available, see the documentation for your network and operating system.

Setting network resource attributes

After you have made network resources available, you must make sure that they possess the necessary attributes to allow all application workstations to access them simultaneously. Each NOS provides a different means of setting the attributes to make network resources shareable.

For example, suppose you want to set the attributes of the OpenEdge installation directory on a network file server so that all workstations can access the OpenEdge files stored there.

OpenEdge is already loaded onto the network file server and is available to the network. The commands used to set resource attributes vary from network to network.

For further information on how to use these or equivalent commands for your network, see the documentation for your network and operating system.

Granting user access rights

After making OpenEdge network resources available and setting resource attributes, you might have to grant access rights to client users and Application Server machines in the network.

Depending on your network, these access rights can include attributes such as read, execute, or open permissions that you must set for each user. See the network documentation for details about how to grant user access rights.

Note: User rights in an Application Server configuration are assigned to the machine where the application server resides, not to the user's client machine.

Remember that an OpenEdge database server can be a user on your network. Like application workstations, it might need user access rights granted to it. If you locate any database files on your network file server, be sure to grant the OpenEdge database server the necessary rights to access the network directory that contains the database.

AdminServer Authorization and Authentication

This appendix addresses additional AdminServer-related activities you can perform in Windows, as described in the following sections:

- [AdminServer logging details](#)
- [Determine the data logged in the AdminServer log](#)
- [Setting authentication option to start servers administered by the AdminServer](#)

Note: The procedures to establish AdminServer authorization options are located in the Windows online help system under these topic titles: “Establishing AdminServer Authorization Options during the Installation” and “Selecting the Authorization Feature when Starting the AdminServer.”

AdminServer logging details

There are logging entries that are specifically related to user authentication and authorization. This section identifies the log format and describes the information that it can contain.

Log format

The log lists both successful and failed operations in the following format:

```
[date][level] ["security"]  UserName:UserSuppliedPwd:GroupInfo:Text
```

Log contents

The following describes the fields in the security entry:

- **Date** — The existing Logging tool automatically inserts the current date using the existing AdminServer log format.
- **Level** — The possible levels are 1 through 5, in compliance with the existing AdminServer log conventions. The security entry will use only the following levels:
 - **0** indicates an internal error.
 - **2** indicates an error condition and explains why the client was not authenticated or authorized.
 - **3** indicates success and is used for tracking purposes.
- **“security”** — This is a text constant that Progress specifies in order to simply log file scanning tools, so that an automated parser can easily identify security events.
- **UserName** — This field contains the user account being authenticated to the AdminServer. This field might indicate “no-user” if the authentication and/or authorization operation failed before the authentication portion could take place. In Windows systems only, the *UserName* might be in the form *[domain\]UserName* where domain is the result of an account lookup operation when the user has not specified a fully qualified user account.

- **UserSuppliedPwd** — This field indicates whether the password being validated for the user account is one of the three following possible conditions:
 - **Y** indicates that the password is supplied by the user.
 - **N** indicates that the password supplied is by the single sign-on password generator.
 - **X** indicates that the password has not yet been validated.
- **GroupInfo** — This field contains group authorization information. When the AdminServer initializes, it validates that a minimum of one group is accessible before allowing startup. In this instance, the field will contain the list of available groups and unavailable groups. Unavailable groups are identified within enclosing braces.

The following example shows the format of **GroupInfo**:

group, group...;{unavailablegroup,unavailablegroup...}

In Windows only, the list of available groups might have the Windows domain prefixed in square brackets to indicate where the group name lookup operation found the entry.

When a security entry is made for an authentication or authorization operation, it can contain:

- **No Group Checking** — This indicates that the AdminServer started without the -admingroup option and no group authorization took place.
 - **GroupName** — This indicates that a single group name was successfully authorized for the user with a success message logged.
 - **GroupNames** — This indicates the group names that the user failed to be authorized in when the failure message was logged.
- **Text** — This field contains one of the messages that further explains the success or failure. The possible text messages follow:
 - User is not authenticated.
 - User is authenticated and authorized.
 - User is not authorized.
 - Failed to find the admingroup(s).

- Failed to find the admingroup, not a valid group list.
- Failed to find the admingroup, please provide a valid group list.
- User password is not valid.
- System generated password has expired.
- Error, system generated password is not valid, user and host are valid.
- Valid group list.

The default behavior for logging is that both success and failure events will be logged.

Determine the data logged in the AdminServer log

There is an AdminServer command-line option for JVMARGS that is called `DLogLevelSecurity`, that, when set, determines the type of logging that the AdminServer log file captures. The syntax for JVMARGS is as follows:

Syntax

```
JVMARGS="$JVMARGS -DLogLevelSecurity={2 | 3}"
```

`DLogLevelSecurity=2`

Stops successful logins from being logged.

`DLogLevelSecurity=3`

Logs failures and successes.

Setting authentication option to start servers administered by the AdminServer

You can require that when users are starting servers of the AdminServer (AppServer, Adapter for SonicMQ, and WebSpeed) the `ubroker.properties` file must provide a valid username and password. This authentication for starting the AppServer, WebSpeed, and Adapter for SonicMQ uses the `ubroker.properties` file hierarchy to find usernames and passwords. The Progress Explorer password field can be set to supply the username's password.

The command-line option that tells the AppServer, WebSpeed, and Adapter for SonicMQ to require a username and password from the `ubroker.properties` file is Require Username (`-requireusername`). You can run `OpenEdge-install-dir\bin\genpassword`. This gives the user an obfuscated password that the user can enter into the Progress Explorer.

The Require Username syntax is as follows:

Syntax

<code>-requireusername</code>

Index

Numbers

4GL Development System
 components during a Complete
 installation 12–5
 components installed during Complete
 installation 13–5

A

Access rights
 networks F–18

Accessing a server behind a firewall 11–19

Additional resources for most current
 system requirements
 Product Availability Guide 1–2
 Release Notes 1–2

Addressing
 OpenEdge brokers/servers 11–16
 OpenEdge hosts 11–16

AdminServer
 additional considerations 10–22
 authorization options 7–13
 changing the default port 10–19
 changing the default startup setting
 10–20
 considerations 7–13

 defined 10–5, 10–9
 hosts C–5
 logging details related to authentication
 and authorization G–2 to G–4
 ports C–5
 querying 10–21
 setting authentication option to start
 servers G–5
 starting 10–16
 stopping 10–18
 understanding and using 10–16 to 10–22

AdminServer plugins.properties file
 defined 10–8

After-image writers (AIW) F–4

alias.pem files 9–6

alias.pk1 files 9–6

alias.pk10 files 9–6

Application Server
 Components of Complete installation
 12–16, 12–19

Application services E–3

Application workstations F–5

Applications
 starting on networks 11–16

- AppServer Internet Adapter
 - components installed during Complete installation 13–9
 - components of Complete installation 12–9
- AppServer Maximum Maintained Prestart Counter (-Mms) parameter 11–15
- AppServer Maximum Prestart Counter (-Ms) parameter 11–15
- Arbitrary sum weight factors E–6
- Asynchronous page writers (APWs) F–4

B

- Background mode
 - multi-user OpenEdge 11–22, 11–23
 - single-user OpenEdge 11–7
- Batch mode
 - defined (UNIX) 5–13
 - defined (Windows) 4–31
 - multi-user OpenEdge 11–22, 11–23
 - oesetup.ini (Windows) 4–31, 5–13
 - single-user OpenEdge 11–12
- Batch startup commands
 - single-user 11–4, 11–10
- Before-image Filename (-g) startup parameter F–16
- Before-image writers F–4
- BPRO utility 11–7, 11–12
- Broadcast request to NameServer E–9
- Brokers
 - addressing 11–16
 - defined F–4
 - multi-broker access 11–18
 - starting 11–16
 - servers for remote users 11–7, 11–13

C

- Certificate store
 - OpenEdge
 - installing trust CA/root certificates 9–9
 - management 9–9
 - managing 7–16, 8–16, 9–1
- certutil
 - syntax and operations 9–11
- certutil tool
 - purpose 9–10
- Changing the AdminServer default port 10–19
- Client Networking
 - components of Complete installation 12–10
- Client Networking, components installed during Complete installation 13–10
- Client startup parameters
 - networks 11–14
- Client/Server configurations F–5
 - samples F–7
 - terminology F–5
- Client/Server OpenEdge
 - addressing 11–16
 - broker/server addressing 11–16
 - execution 11–13
 - starting clients and servers 11–13
- Clients
 - defined F–5
 - OpenEdge
 - installing trusted CA/root certificates 9–9
 - SSL certificate store management 9–9
 - starting
 - TCP/IP on UNIX 11–17
- Command-line utilities
 - defined 10–6

- Commands, mount 5–4
 - Compatibility
 - ABL clients and database servers (on UNIX) 2–8
 - Complete installation 3–6
 - Completing OpenEdge installation in Windows 4–4
 - Components
 - used to calculate memory needs 6–15
 - Components of Complete installation
 - OpenEdge Adapter for Sonic ESB 12–15
 - Personal RDBMS 12–45
 - Components of Complete installation in Windows 12–1 to 12–72
 - 4GL Development 12–5
 - Application Server - Basic 12–16
 - Application Server Enterprise 12–19
 - AppServer Internet Adapter 12–9
 - Client Networking 12–10
 - Enterprise RDBMS 12–41
 - NameServer Load Balancer 12–15
 - ODBC DataServer 12–31
 - OpenEdge Adapter for Sonic ESB 12–15
 - OpenEdge Application Server Enterprise 12–19
 - OpenEdge Architect 12–23
 - OpenEdge Development Server 13–24
 - OpenEdge Replication 12–49
 - OpenEdge Replication Plus 12–50
 - OpenEdge Studio 12–52
 - Oracle DataServer 12–34
 - Personal RDBMS 12–45
 - Query/Results 12–61
 - Translation Manager 12–63
 - Visual Translator 12–64
 - Web Services Adapter 12–66
 - WebSpeed Messenger 12–66
 - WebSpeed Workshop 12–67
 - Workgroup RDBMS 12–57
 - Components on UNIX
 - installed during a Complete installation
 - 4GL Development System 13–5
 - AppServer 13–16
 - AppServer Internet Adapter 13–9
 - Client Networking 13–10
 - Enterprise Database 13–28
 - NameServer 13–14
 - NameServer Load Balancer 13–15
 - OpenEdge Replication 13–13
 - OpenEdge Replication Plus 13–13
 - Oracle DataServer 13–21
 - Personal Database 13–32
 - Secure AppServer 13–21
 - WebSpeed Messenger 13–41
 - Workgroup Database 13–35
 - Configuration XML file
 - editing manually 4–15
 - Configurations
 - client/server F–5
 - displaying information 6–9
 - networks F–9
 - shared memory F–2
 - Configuring
 - NameServers
 - using Progress Explorer E–18
 - OpenEdge on an NOS F–17
 - conmgr.properties file
 - defined 10–7
 - Connection parameters
 - icfdb 4–5
 - Connection-level fault tolerance E–3
 - two techniques E–8
 - understanding E–7
 - Control number
 - displaying 6–6
 - Custom installation 3–6
- ## D
- d9855a82.0 file 9–9
 - Data Administration Tool
 - creating a database 7–14

- Data Dictionary
 - creating a database 7–14
- Database server machines
 - defined F–5
 - multi-process
 - defined F–7
 - single-process
 - defined F–6
- Databases
 - monitoring F–4
 - startup 11–16
- DataSource for ODBC
 - components of Complete installation 12–31
- DataSource for Oracle
 - components of Complete installation 12–34
- Default services E–4
 - defined E–4
- Default SSL identity
 - Progress CA root digital certificate 9–9
- DefaultFont environment variable
 - international progress.ini files D–15
- Delta files
 - defined 10–26
- Disk space requirements
 - OpenEdge products installed in Windows 3–8
 - OpenEdge products installed on UNIX 3–10
- DLC
 - and OpenEdge-install-dir 3–18, 3–25
 - environment variable, UNIX 8–4
 - setting the environment variable (Windows) 7–6
- Downloading executables for
 - heterogeneous environments 4–30, 5–13
- Dynamics Configuration Utility (DCU)
 - running 4–5

E

- Enterprise Database, components installed during Complete installation 13–28
- Enterprise RDBMS
 - components of Complete installation 12–41
- Environment variables
 - Java (Windows) 7–4
 - PATH, UNIX 8–4
 - setting (Windows) 7–3
 - UNIX 8–2, 8–4
 - Windows 7–5
- Error code parameters 6–21
- Error messages
 - related to installation 6–23
- European Numeric Format (-E) D–13
- Expiration date
 - displaying 6–6

F

- Fail-over weight factor E–7
- File descriptors
 - UNIX requirements for WebSpeed 2–7
- File types
 - supported by OpenEdge 3–21
- Files
 - conmgr.properties C–4
 - icf.ini 4–15
 - icf.pf 4–15
 - icfconfig.xml 4–15
 - network sharing F–18
 - protocols, *See* protocols file
 - services, *See* services file
 - startdbs.bat 4–15
 - stopdbs.bat 4–15
- Formulas, for calculating memory requirements 6–17

G

genpassword tool
 example 9–8
 purpose 9–8

H

-H startup parameter
 client connection 11–14
 server startup 11–14
 specifying 11–16

Host addressing 11–16

Host Name (-H) parameter
 client connection 11–14
 server startup 11–14
 specifying 11–16

Host request to NameServer E–9

HP Tru64
 Java requirements for 2–5
 preinstallation tasks for 3–23

HP-UX
 Java requirements for 2–4
 modifying JDKHOME value for 8–9

I

IBM AIX
 Java requirements for 2–4

icf.ini file
 editing manually 4–15

icf.pf file
 editing manually 4–15

ICFDB database 4–5
 creating 4–5

Icons
 used by OpenEdge 3–22

Installation

 directory structure in Windows 3–17
 directory structure on UNIX 3–25
 displaying configuration information 6–9
 displaying date 6–6

Installation activities

 additional tasks in Windows 4–25

Installation considerations

 UNIX-specific 3–23
 Windows-specific 3–12

Installation media

 loading 4–2
 types of 4–2

Installation method

 defined 3–5
 online, interactive 3–5
 silent (or batch mode) 3–5

Installation type

 Complete option 3–6
 Custom option 3–6
 defined 3–6

Installing and managing keys and digital

 certificates
 certutil C–15
 genpassword C–17
 mkhashfile C–17
 pkiutil C–18

Installing OpenEdge 4–1 to 4–31

 and performing postinstallation tasks (on UNIX) 5–22
 choosing an installation type when 13–4
 finishing 4–4
 in a heterogeneous environment 4–30
 in a heterogeneous environment (on UNIX) 5–13
 troubleshooting when 6–22 to 6–25
 viewing registry information 4–29
 for ODBC driver 4–29

Installing OpenEdge in Windows

 post-installation tasks 4–41

Installing OpenEdge products to a current installation 4–26

Interactive mode
multi-user OpenEdge 11–20

Interactive mode, multi-user OpenEdge 11–21

International
databases D–5
parameter files D–12

J

Java
system requirements (in Windows) 1–2

Java Development Kit (JDK)
environment variables (Windows) 7–4

Java requirements on UNIX 2–2
JDK 2–2
JRE 2–3

Java requirements specific to each UNIX platform 2–5

Java Runtime Environment (JRE)
environment variables (Windows) 7–4

JavaTools.properties file
defined 10–8

JDKHOME
setting (Windows) 7–4

JDKHOME, setting (UNIX) 8–9

K

Kernel
configuration 6–20
reconfiguration parameters 6–21

Key store
OpenEdge
backup subdirectory 9–7

content and structure 9–6
creating and managing entries 9–3
management 9–2
managing 7–16, 8–16, 9–1
policy subdirectory 9–6
requests subdirectory 9–6
specifying entry password 9–4

L

LAN OpenEdge
startup parameters 11–13

Language
support D–1

License
information 2–12

License Addendum 3–3

License Update utility
changes to accommodate license updates 6–2
defined 6–2
updating a PSDN subscription renewal 6–2
updating an evaluation license to a non-evaluation license 6–2

Licenses
displaying current information 6–5
managing 6–9
user counts 6–9

Load balancing
arbitrary sums E–6
fail-over E–7
percentage-based E–6
priority weight factors E–5

Loading
remote databases 4–5

Local NameServer instances E–17

Location transparency E–2

Log (.lg) files
commands to removing entries 6–26

M

-m2 startup parameter 11–14

-m3 startup parameter 11–14

-Ma startup parameter 11–14

Machine addressing 11–16

Machine Class
displaying 6–6

Maintaining two versions of OpenEdge and
Progress (Windows) 7–15

Manual Server (-m2) startup parameter
11–14

Maximum Clients per Server (-Ma)
parameter
server startup 11–14

Maximum Dynamic Server (-Maxport)
parameter
server startup 11–14

Maximum Servers (-Mn) parameter
server startup 11–15

Maximum Servers per Broker (-Mpb)
startup parameter
server startup 11–15

Maximum Servers per Protocol (-Mp)
parameter
server startup 11–15

-Maxport startup parameter 11–14

MBPRO command 11–22, 11–23

Memory
and semaphore parameter settings 6–20,
6–21
and setting swap space 6–19
calculating requirements for 6–14, 6–16
formulas to calculate requirements for
6–17
multi-user requirements for 6–17

reducing usage 6–19
shared 6–20
parameter settings 6–20, 6–21
single-user requirements for 6–17

Mergeprop utility
actions 10–31
defined 10–6, 10–26
delta file 10–30
example uses 10–32
overview 10–26
property files managed by 10–30
syntax 10–27
target file 10–29

Maximum Maintained Prestart Counter for
AppServer (-Mms) parameter 11–15

Message Buffer Size (-Mm) parameter
client connection 11–14
server startup 11–15

-Mi startup parameter
server startup 11–14

Minimum Clients per Server (-Ma)
parameter
server startup 11–14

Minimum Dynamic Server (-Minport)
parameter
server startup 11–14

Minimum operating system level, by
platform 2–8

-Minport startup parameter
server startup 11–14

mkhashfile tool
purpose 9–10
syntax and operations 9–12

-Mm startup parameter
client connection 11–14
server startup 11–15

-Mms startup parameter 11–15

-Mn startup parameter
server startup 11–15

- Modifying
 - an OpenEdge installation 4–43
- Mount commands 5–4
- Mp startup parameter
 - server startup 11–15
- Mpb startup parameter
 - server startup 11–15
- MPRO command 11–17
- MPRO utility 11–20, 11–21
- Ms startup parameter
 - server startup 11–15
- Multi-threaded servers
 - F–5
- Multi-user memory requirements 6–17
- Multi-user OpenEdge 11–8, 11–20, 11–23
 - commands to start
 - batch mode 11–22
 - interactive mode 11–20
 - startup commands 11–8
 - batch mode 11–22
 - broker 11–8
 - interactive mode 11–20
 - server 11–7
 - Windows service 11–7
 - commands to start
 - batch mode 11–23
 - interactive mode 11–21
 - starting the server 11–13
 - startup commands
 - batch mode 11–23
 - broker 11–13
 - interactive mode 11–21
 - server 11–13
- Multi-user OpenEdge startup commands
 - server 11–8
- Multi-user OpenEdge
 - starting the server 11–7

N

- N startup parameter
 - client connection 11–14
 - defaults 11–15
 - server startup 11–15
 - specifying 11–15
- NameServer
 - accessing its port and host E–4
 - broadcast address E–12
 - client connections E–17
 - components installed during Complete installation 13–14
 - configuring
 - using Progress Explorer E–18
 - configuring communications E–4
 - downloading executables E–16
 - editing the services file E–5
 - instance configuration E–17
 - neighbors E–8
 - example E–14
 - using E–13
 - replication E–8
 - example E–10
 - using E–10
 - starting instances
 - using Progress Explorer E–19
 - UDP broadcasting E–9
- NameServer Load Balancer Option
 - components of Complete installation 12–15
- NameServer Load Balancer, components installed during Complete installation 13–15
- NameServers
 - defined E–2
- Neighbor NameServers E–8
- NetSetup
 - running the Shared Network Installation Utility 4–53
 - shortcuts 4–53

- Network file servers
 - defined F-6
 - Network operating system
 - access rights F-18
 - accessing resources F-17
 - OpenEdge startup 11-13
 - Network startup parameters 11-13
 - Network Type (-N) parameter
 - client connection 11-14
 - defaults 11-15
 - protocol support F-9
 - server startup 11-15
 - specifying 11-15
 - Network Version (-Nv) startup parameter
 - client connection 11-14
 - Networks
 - access rights F-18
 - accessing resources F-17
 - applications
 - starting 11-16
 - client startup parameters 11-14
 - client/server
 - startup 11-16
 - configurations F-9
 - database server machines F-5
 - file server 4-51, F-6
 - applications F-11, F-13
 - as database server machine F-12
 - file servers F-6
 - OpenEdge startup parameters 11-13
 - PC LAN permissions F-18
 - resource sharing F-18
 - server startup parameters 11-14
 - types F-9
 - types defined F-9
 - Nv startup parameter 11-14
- O**
- ODBC DataServer
 - components of Complete installation 12-31
 - Online, interactive installation method 3-5
 - OpenEdge
 - broker/server addressing 11-16
 - client/server startup 11-13
 - configuration file (Progress.cfg) 6-23
 - configuring on an NOS F-17
 - icons 3-22
 - installing 4-1 to 4-31
 - installing (UNIX) 5-1 to 5-25
 - integration with Windows Explorer 3-21 to 3-22
 - messages, *See* PROMSGS
 - network addressing 11-16
 - network startup parameters 11-13
 - starting multi-user 11-23
 - batch mode 11-22, 11-23
 - interactive mode 11-20, 11-21
 - starting single-user
 - batch mode 11-7, 11-11
 - interactive mode 11-6, 11-11
 - supported file types 3-21
 - supported platforms (UNIX) 2-8
 - OpenEdge Adapter for Sonic ESB
 - components installed during Complete installation 13-15
 - components of Complete installation 12-15
 - OpenEdge and Progress
 - maintaining two versions (Windows) 7-15
 - OpenEdge Application Server
 - components of Complete installation 12-19
 - OpenEdge Application Server Basic
 - components of Complete installation 12-16
 - OpenEdge AppServer, components
 - installed during Complete installation 13-16
 - OpenEdge Architect 3-32
 - components of Complete installation 12-23

- OpenEdge client/server
 - network file server
 - application files F-11, F-13
 - network file server and
 - as database server machine F-12
 - running on a LAN 11-13
 - startup parameters 11-13
 - with AppServer F-14
- OpenEdge configuration file (Progress.cfg),
 - altered or missing 6-23
- OpenEdge Database Monitor utility (PROMON) F-4
- OpenEdge Development Server
 - components of Complete installation 12-36, 13-24
- OpenEdge directory structure
 - in Windows 3-18
 - on UNIX 3-25
- OpenEdge installation
 - TCP/IP networks F-15
- OpenEdge installation media
 - requirements 1-4
- OpenEdge key and certificate stores
 - introduction 7-16
- OpenEdge product
 - details about components and subcomponents 13-1
- OpenEdge product box 3-3
- OpenEdge Replication 13-13
 - components installed during Complete installation 12-49
- OpenEdge Replication Plus
 - components installed during Complete installation 12-50, 13-13
- OpenEdge sample configurations
 - network file servers
 - application files F-11, F-13
 - as database server machine F-12
 - with AppServer F-14
- OpenEdge Silent Installation Utility
 - syntax (UNIX) 5-19
- OpenEdge Silent Installation utility
 - syntax (Windows) 4-37
- OpenEdge SSL
 - certutil tool 9-10
 - clients
 - certificate store management 9-9
 - installing root CA certificates 9-9
 - key and certificate stores 7-16, 8-16, 9-1
 - key store content 9-6
 - key store entry password 9-4
 - mkhashfile tool 9-10
 - pkiutil tool 9-2
 - servers
 - establishing identity 9-3
 - key store management 9-2
- OpenEdge Studio
 - components of Complete installation 12-23
- OpenEdge Watchdog utility (PROWDG) F-4
- OpenEdge-install-dir
 - and DLC 3-18, 3-25
- Oracle DataServer
 - components of Complete installation 12-34
- Oracle DataServer, components installed during Complete installation 13-21

P

- Parameter file
 - typical startup parameters D-12, D-13
- Parameters
 - icfdb startup 4-5
- Parsing XML files 4-5
- Password for OpenEdge key store entries 9-4

- PATH, environment variable, UNIX 8–4
- Percentage-based weight factors E–6
- Permissions
 - networks F–18
- Personal Database, components installed during Complete installation 13–32
- Personal RDBMS
 - components of Complete installation in Windows 12–45
- pkiutil tool
 - purpose 9–2
 - syntax and operations 9–5
 - using 9–3
- Planning your installation 3–3
- Platform support for OpenEdge Management 3.1B 1–14
- Platforms, supported (UNIX) 2–8
- Port Number
 - displaying 6–6
- Post-installation tasks (UNIX) 5–22
- Post-installation tasks (Windows) 4–41
- Preinstallation checklist
 - for UNIX B–1
 - for Windows A–1
- Preinstallation documentation resources 3–3
 - checklists 3–3
 - License Addendum 3–3
 - OpenEdge Installation online help (Windows) 3–4
 - product component and subcomponent details 3–4
 - Release Notes 3–4
- Preinstallation tasks 3–2
 - for HP Tru64, AIX, Unixware, and Linux 3–23
 - for SQL install 3–14
 - on a shared network 4–51, 4–53, 4–57
- Preinstallation tasks overview 3–2
- PrinterFont environment variable
 - international progress.ini files D–15
- Priority weight factor
 - load balancing E–5
- PRO utility 11–6, 11–11
- PROAB section in progress.ini 4–4
- Procedure Editor startup commands
 - UNIX, Windows 11–4
- Procedure Editor startup commands, UNIX, Windows 11–10
- PROCFG, environment variable, UNIX 8–5
- PROCONV, environment variable, UNIX 8–5
- PRODB
 - creating a database 7–14
- PRODB utility 4–5
- Product Availability Guide 2–2
 - accessing 1–2
- Product configurations
 - ubroker.properties file 10–38
- Product name
 - displaying 6–6
- Product update utility 6–3
- Product Update utility, *See also* License Update utility
- PROEXE, environment variable, UNIX 8–5
- Program item properties
 - setting 7–11
- Progress Download Center
 - accessing 3–3

- Progress Dynamics
 - Completing the DCU wizard 4–7
- Progress Explorer
 - using 10–24
- Progress Explorer configuration tool
 - defined 10–5
 - launching 10–24
 - using 10–23
- Progress Explorer Framework
 - additional considerations 10–9
 - defined 10–2
 - element and description overview 10–3
 - elements of 10–5
 - graphical overview 10–4
 - OpenEdge database 10–3
 - OpenEdge products that support broker functionality 10–3
 - Unified Broker 10–3
 - common elements 10–10
 - configuring and starting 10–13
 - default sample brokers 10–11
 - details 10–9
- Progress Explorer, *See* Progress Explorer configuration tool
- Progress Version 9 3–24
- progress.cfg file 6–23
- progress.ini file
 - editing D–14
 - Windows 7–4
- PROLOAD, environment variable, UNIX 8–5
- PROLOG
 - removing log entries 6–26
- PROMON utility (OpenEdge Database Monitor) F–4
- PROMSGS D–6
 - defined D–2
 - environment variable D–15
 - environment variable, UNIX 8–5
 - messages D–6
 - removing prior to a reinstall 3–16
 - shell integration 3–22
 - translations D–2
- PROPATH, environment variable, UNIX 8–7
- Properties file
 - general syntax 10–35
 - information added by OpenEdge 3–22
 - ubroker.properties 10–38
- PROSERVE
 - command 11–16
 - utility 11–13, 11–18
- PROSRV, environment variable, UNIX 8–8
- PROSTART, environment variable, UNIX 8–8
- PROTERMCAP, environment variable, UNIX 8–8
- Protocols file E–4
- PROWDog utility (OpenEdge Watchdog) F–4
- pscca.cer file 9–9
- psempi.cnf file 9–6

Q

- Query/Results
 - components of Complete installation 12–61

R

- Regional support
 - installing PROMSGS D–5
- Registry
 - editing D–14
 - Windows 7–4

- Release Notes 2–2
 - accessing 1–2
- Remote database
 - loading 4–5
- Remote NameServer instances E–17
- Removing log file entries
 - PROLOG 6–26
- Removing OpenEdge
 - manually (Windows) 4–46
- Replicated NameServers E–8
- Repository
 - creating 4–5
- Required third-party applications for
 - Windows
 - DataDirect ODBC branded drivers 1–11
 - DataDirect SQL drivers 1–11
 - Microsoft Management Console 1–10
- Required third-party applications in
 - Windows 1–10
- Resources
 - network sharing F–18
- Root CA digital certificate
 - Progress CA 9–9
- Running OpenEdge
 - in Windows 7–15
- Running the Shared Network Installation
 - Utility (NetSetup) 4–53
- S**
- S startup parameter 11–16
 - client connection 11–14
 - server startup 11–15
- Saving an existing OpenEdge or Progress
 - installation 3–14, 3–17, 3–24
- Secondary Login Broker (-m3) startup
 - parameter
 - server startup 11–14
- Secure AppServer
 - components installed during Complete
 - installation 13–18
- Self-service clients
 - defined F–2
- Semaphore parameter settings 6–20, 6–21
- Serial number
 - displaying 6–6
- Server access
 - behind a firewall 11–19
- Server compatibility
 - .dotNet Client details (Windows) 1–8
 - ABL clients and database servers
 - (Windows) 1–5
 - AppServer (Windows) 1–7
 - Dataserver (Windows) 1–6
 - SQL Database Server (Windows) 1–7
 - Transaction server (Windows) 1–9
- Server compatibility in Windows 1–5
- Server startup parameters
 - networks 11–14
- Server-level fault tolerance E–2
 - understanding E–7
- Servers
 - addressing 11–16
 - defined F–5
 - network F–5
 - OpenEdge
 - establishing SSL identity 9–3
 - SSL key store management 9–2
 - starting 11–16
- Service Name (-S) startup parameter 11–16
 - client connection 11–14
 - server startup 11–15
- Services file E–5

- Setting
 - environment variables (Windows) 7-3
 - Java environment variables (Windows) 7-4
 - JDKHOME (Windows) 7-4
- Setting environment variables for SQL (UNIX) 8-10
- Setting up the OpenEdge environment
 - [Startup] section 7-5, 7-6, 7-7, 7-8
 - [WinChar Startup] section 7-5, 7-6, 7-8
 - DLC variable (Windows) 7-6
 - environment variables (Windows) 7-3
 - Java environment variables (Windows) 7-4
 - program item properties (Windows) 7-11
- Setting up the OpenEdge environment (Windows) 7-1
- Shared memory 6-20, F-2
 - architecture F-3
 - configurations F-2
 - parameter settings 6-20, 6-21
- Shared network
 - preinstallation tasks when installing on a 4-51, 4-53, 4-57
- Shared Network Installation Utility
 - defined (Windows) 4-50
- Shared Network Installation utility 3-7, 4-50 to 4-57
- Sharing an OpenEdge installation on a network 4-50 to 4-57
 - primary tasks 4-50
 - running the Silent installation option 4-58
 - setting up the shared network 4-52
- Shortcut menu
 - using with OpenEdge 3-22
- Show Configuration (SHOWCFG) utility 6-3, 6-8, 6-9
- Silent install, *See also* Batch mode
- Silent Installation
 - defined 3-5
- Silent installation in Windows
 - data input options 4-32
 - log file details 4-38
 - optional activities
 - creating data input 4-40
 - manually modifying data input 4-41
 - overview 4-31
 - response.ini 4-33
 - running
 - commands 4-37
- Silent installation on UNIX
 - data input options 5-14
 - log file details 5-20
 - optional activities
 - creating data input 5-21
 - manually modifying data input 5-21
 - overview 5-13
 - response.ini 5-15
 - running
 - commands 5-18
- Single-threaded servers
 - F-5
- Single-user memory requirements 6-17
- Single-user OpenEdge
 - startup commands
 - background mode 11-7
 - batch mode 11-7
 - interactive mode 11-6
- Single-user OpenEdge startup commands
 - batch mode 11-11
- Single-user OpenEdge, startup commands
 - interactive mode 11-12
- Size increments for increasing startup parameters 6-16
- SQL Server
 - components of Complete installation 12-29

- SQL-92, system requirements (UNIX) 2–5
 - startdbs.bat file
 - editing manually 4–15
 - Starting
 - clients and servers 11–13
 - network applications 11–16
 - Starting OpenEdge
 - multi-user 11–23
 - batch mode 11–21
 - batch mode in Windows 11–22
 - interactive mode in windows 11–20
 - interactive mode on UNIX 11–21
 - Windows service 11–8
 - with a server 11–7, 11–13
 - multi-user on UNIX
 - single-user
 - batch mode 11–7, 11–11
 - interactive mode 11–6, 11–11
 - Startup commands
 - summary 11–2
 - Startup commands, summary 11–9
 - Startup parameters
 - client 11–14
 - deployed applications D–12
 - icfdb 4–5
 - server 11–14
 - size increments for increasing 6–16
 - Startup scripts, tailoring 6–24, 6–25
 - stopdbs.bat file 4–15
 - Supported, platforms on UNIX 2–8
 - Swap space 6–19
 - System requirements
 - for building OpenEdge applications (UNIX) 2–6
 - for running OpenEdge applications (UNIX) 2–7
 - Java
 - HP Tru64 2–5
 - HP-UX 2–4
 - HP-UX Itanium 2–5
 - IBM AIX 2–4
 - Linux 2–5
 - RedHat Enterprise Linux 2.1 2–5
 - Solaris SPARC 2–5
 - SQL-92 (on UNIX) 2–5
 - Java (on UNIX) 2–2
 - Java (Windows) 1–2
 - SuSE Linux Enterprise Server 8 2–5
 - Windows 2000 1–2, 1–3
 - Windows 2003 1–2, 1–3
 - Windows XP Professional 1–2, 1–3
 - System requirements for Windows
 - required third-party applications 1–10
 - System requirements on UNIX and Linux 2–2
- ## T
- Table 12–23
 - Taget files
 - defined 10–29
 - Tailoring startup scripts 6–24
 - Target files
 - defined 10–26
 - TCP/IP clients
 - MPRO command 11–17
 - starting in Windows 11–17
 - starting on UNIX 11–17
 - TCP/IP networks
 - installing OpenEdge F–15
 - network files F–17
 - preparing for OpenEdge F–17
 - without file server F–16
 - TERM, environment variable, UNIX 8–6, 8–8
 - Terminal identifiers 8–14
 - Translation Manager
 - components of Complete installation 12–63

Troubleshooting an installation 6–22 to 6–25

U

ubroker.properties file 10–38
 defined 10–7
 products supported 10–41
 structure 10–40

UDP broadcasting E–9
 performance implications E–15

ULIMIT

 WebSpeed Agents requirement (on UNIX) 2–7

Unified Broker

 additional characteristics 10–12
 and Name Server relationship E–3
 application services E–3
 clients 10–39
 common elements of 10–10
 default sample broker for each Unified Broker product 10–11
 installation prerequisites 10–39
 NameServer instances E–17
 order of configuration E–16
 properties file
 guidelines for editing 10–42
 using default sample brokers 10–11

Unified Broker products

 administering and configuring C–2
 ASBMAN C–2
 DBMAN C–4
 mergeprop C–5
 NSMAN C–8
 PROADSV C–10
 that support broker functionality 10–3
 ubroker.properties file 10–3
 WTBMAN C–13

Uninstalling OpenEdge or Progress 4–43 to 4–49

 Uninstall or Add/Remove Programs Utility 4–43 to 4–45

UNIX

 default network type 11–15
 environment variables 8–2, 8–4 to 8–6
 starting network clients 11–17

Upgrading

 an OpenEdge installation 4–43 to 4–49

User licenses, *See* Licenses

Utilities

 BPRO 11–7, 11–12
 MBPRO 11–22, 11–23
 mergeprop 10–26 to 10–37
 MPRO 11–20, 11–21
 PRO 11–6, 11–11
 ProControl 11–23
 product update 6–3
 PROMON F–4
 PROSERVE 11–18
 PROWDOG F–4

V

Version number

 displaying 6–6

Viewing registry information 4–29

Visual Translator

 components of Complete installation 12–64

W

Watchdog utility F–4

Web browser

 requirements (Windows) 1–4

Web development
set up 4–25

Web server
requirements (Windows) 1–4

Web Services Adapter
components of Complete installation
12–66

WebSpeed
Configuration choices (Windows) 3–32

WebSpeed Messenger
components of Complete installation
12–66

WebSpeed Messenger, components
installed during Complete installation
13–41

WebSpeed Workshop
components of Complete installation
12–67

Windows Explorer
integrating with OpenEdge 3–21 to 3–22

Workgroup Database, components installed
during Complete installation 13–35

Workgroup RDBMS
components of Complete installation
12–57

Working directory
defined 3–13

Working Path
DCU 4–5

X

XML files
parsing 4–5

