Document Revision History

This document has been revised since it was first published. Refer to the table below for details. Also, change bars appear next to the changed areas.

<table>
<thead>
<tr>
<th>Published Date</th>
<th>Description of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2009</td>
<td>Initial publication for OpenEdge® Release 10.2B. See <em>OpenEdge Getting Started: New and Revised Features</em> for information on the new and revised features documented for 10.2B.</td>
</tr>
<tr>
<td>February 2010</td>
<td>• Increased the default value and maximum value for the Nested Blocks (-nb) client startup parameter. For more information, see the “Nested Blocks (-nb)” section on page 1–4.</td>
</tr>
</tbody>
</table>
## Contents

**Preface**  ..........................................................  Preface–1

1. **What’s New in 10.2B.**  ........................................  1–1
   Installation  .........................................................  1–2
   OpenEdge startup parameters  .....................................  1–3
      Blocks in Alternate Buffer Pool (-B2)  ..........................  1–3
      Nested Blocks (-nb)  .............................................  1–4
      Prompt for Passphrase (-Passphrase)  ..........................  1–5
      Re-usable Objects Cache (-reusableObjects)  .................  1–5
   OpenEdge Architect  ..............................................  1–7
      Upgrade to Eclipse 3.4.2  .....................................  1–7
      File layout changes  ............................................  1–8
      New video demonstrations  ......................................  1–8
      Upgrade of OpenEdge Ultra Controls for .NET  ...............  1–9
      Update Assembly References Tool.  ...........................  1–9
      Support for ABL language improvements  .......................  1–9
      ABL Editor context-filtered code assistance  ..................  1–10
      Improved ProBindingSource Designer  ..........................  1–10
      Global Toolbox option  .........................................  1–10
      Shared AVM  .....................................................  1–10
      Performance improvements  .....................................  1–11
      AppServer support  .............................................  1–11
      Class Browser enhancements.  ................................  1–12
      Support for AppBuilder shortcut keys  .........................  1–12
   OpenEdge GUI for .NET  ..........................................  1–13
      New ABL support for .NET features  ............................  1–13
      New ProBindingSource properties  ..............................  1–13
   ABL enhancements  ..............................................  1–14
      Re-use of ABL class objects  ..................................  1–14
      Abstract classes  ..............................................  1–15
      Class events  ...................................................  1–16
      .NET generic types  ............................................  1–17
      ABL Compiler XREF output  .....................................  1–18
      ABL logging  .....................................................  1–18
      Enhanced reflection, dynamic instantiation, and invocation in
      object-oriented ABL  ............................................  1–19
Using either the unglue or reglue script ........................................... 2–9
Using Proenv in place of Set OpenEdge Management Environment .. 2–10

3. **Release 10.2B Documentation** .................................................. 3–1
   Changes to the documentation set ............................................... 3–2
   Release 10.2B documentation set .............................................. 3–3
   Accessing OpenEdge Documentation ........................................... 3–6
This Preface contains the following sections:

- Purpose
- Audience
- Organization
- Using this manual
- Typographical conventions
- Examples of syntax descriptions
- OpenEdge messages
- Third party acknowledgements
Preface

Purpose

OpenEdge Getting Started: New and Revised Features briefly describes both new features and changes to existing features introduced in OpenEdge® Release 10.2B, and directs you to where you can find more detailed information about these changes in the documentation set. The Release 10.2B documentation set references include:

- **Product manuals** — Identified by the Manual label in tables throughout this manual
- **HTML-based online help** — Identified by the Online help label in tables throughout this manual
- **Web papers** — Identified by the Web paper label in tables throughout this manual

Audience

This guide is primarily intended for OpenEdge application developers and system administrators who are upgrading their license to Release 10.2B from Release 10.2A. It is also a good information source for other existing Release 10.1C users who want to know more about the new and changed features available in Release 10.2B.

Organization

Chapter 1, “What’s New in 10.2B”

Presents brief explanations of new and enhanced features introduced in Release 10.2B, and references places within the documentation set where more detailed feature information can be found.

Chapter 2, “10.2B Feature Comparisons”

Describes how features that changed in OpenEdge Release 10.2B differ from how they functioned in previous releases. The comparisons describe possible changes in the behavior of a tool or application code, and also describe the changes in the documentation.

Chapter 3, “Release 10.2B Documentation”

Provides information about the OpenEdge Release 10.2B documentation set.

Using this manual

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

For the latest documentation updates see the OpenEdge Product Documentation on PSDN: [http://communities.progress.com/pcom/docs/DOC-16074](http://communities.progress.com/pcom/docs/DOC-16074).
References to ABL compiler and run-time features

ABL is both a compiled and an interpreted language that executes in a run-time engine. The documentation refers to this run-time engine as the *ABL Virtual Machine (AVM)*. When the 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 the documentation refers to run-time behavior in an executing ABL program, it specifies the *AVM* as the actor that manages the specified run-time behavior in the program.

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

References to ABL data types

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

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

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

Typographical conventions

This manual uses the following typographical conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>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.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>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.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>A fixed-width font is used in syntax statements, code examples, system output, and filenames.</td>
</tr>
<tr>
<td>Fixed width</td>
<td>A fixed-width font is used in syntax statements, code examples, system output, and filenames.</td>
</tr>
<tr>
<td>Fixed-width italics</td>
<td>Fixed-width italics indicate variables in syntax statements.</td>
</tr>
<tr>
<td>Fixed-width bold</td>
<td>Fixed-width bold indicates variables with special emphasis.</td>
</tr>
<tr>
<td>UPPERCASE fixed width</td>
<td>Uppercase words are ABL keywords. Although these are always shown in uppercase, you can type them in either uppercase or lowercase in a procedure.</td>
</tr>
<tr>
<td>![ ]</td>
<td>This icon (three arrows) introduces a multi-step procedure.</td>
</tr>
<tr>
<td>![ ]</td>
<td>This icon (one arrow) introduces a single-step procedure.</td>
</tr>
<tr>
<td>Period (.) or colon (:)</td>
<td>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.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Large brackets indicate the items within them are optional.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Small brackets are part of ABL.</td>
</tr>
<tr>
<td>{ }</td>
<td>Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.</td>
</tr>
<tr>
<td>{ }</td>
<td>Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>Ellipses indicate repetition: you can choose one or more of the preceding items.</td>
</tr>
</tbody>
</table>
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 NO-LOCK:
    DISPLAY Customer.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 \([ \text{(n)} \)]. and any number of additional expression or SKIP \([ \text{(n)} \)] is allowed:

**Syntax**

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

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

**Syntax**

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

**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**

\[
\text{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**

\[
\text{ASSIGN \{ \[ FRAME frame \] \{ field \[ = \text{expression} \] \}
[ WHEN expression \] \} ...}
\text{\| \{ record \[ \text{EXCEPT field} \ldots \] \}}
\]
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**.
Third party acknowledgements

OpenEdge includes AdventNet - Agent Toolkit licensed from AdventNet, Inc. [http://www.adventnet.com]. All rights to such copyright material rest with AdventNet.

OpenEdge includes ANTLR (Another Tool for Language Recognition) software Copyright © 2003-2006, Terence Parr. All rights reserved. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. 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 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 Concurrent Java software Copyright 1994-2000 Sun Microsystems, Inc. All Rights Reserved. Neither the name of or trademarks of Sun may be used to endorse or promote products including or derived from the Java Software technology without specific prior written permission; and Redistributions of source or binary code must contain the above copyright notice, this notice and the following disclaimers: This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN MICROSYSTEMS, INC. OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN MICROSYSTEMS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

OpenEdge includes DataDirect software Copyright © 1991-2007 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect for JDBC Type 4 driver); Copyright © 1993-2009 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect for ODBC); and Copyright © 1988-2007 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved.
Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect64 for ODBC).

OpenEdge includes DataDirect Connect for ODBC and DataDirect Connect64 for ODBC software, which include ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

OpenEdge includes DataDirect Connect for ODBC and DataDirect Connect64 for ODBC software, which include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). Copyright © 1998-2006 The OpenSSL Project. All rights reserved. And Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

OpenEdge includes DataDirect products for the Microsoft SQL Server database which contain a licensed implementation of the Microsoft TDS Protocol.

OpenEdge includes software authored by David M. Gay. Copyright © 1991, 2000, 2001 by Lucent Technologies (dtoa.c); Copyright © 1991, 1996 by Lucent Technologies (g_fmt.c); and Copyright © 1991 by Lucent Technologies (rnd_prod.s). 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 AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes software authored by David M. Gay. Copyright © 1998-2001 by Lucent Technologies All Rights Reserved (decstrtod.c; strtodg.c); Copyright © 1998, 2000 by Lucent Technologies All Rights Reserved (decstrtol.c; strtord.c); Copyright © 1998 by Lucent Technologies All Rights Reserved (dmisc.c; gdtoa.h; gethex.c; gmisc.c; sum.c); Copyright © 1998, 1999 by Lucent Technologies All Rights Reserved (gdtoa.c; misc.c; smisc.c; ulp.c); Copyright © 1998-2000 by Lucent Technologies All Rights Reserved (gdtoaimp.h); Copyright © 2000 by Lucent Technologies All Rights Reserved (hd_init.c). Full copies of these licenses can be found in the installation directory, in the c:/OpenEdge/licenses folder. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHERWISE TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
OpenEdge includes http package 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 ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.


OpenEdge includes Infragistics NetAdvantage for .NET v2009 Vol 2 Copyright © 1996-2009 Infragistics, Inc. All rights reserved.

OpenEdge includes JSTL software Copyright 1994-2006 Sun Microsystems, Inc. All Rights Reserved. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes OpenSSL software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). Copyright © 1998-2007 The OpenSSL Project. All rights reserved. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 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 Quartz Enterprise Job Scheduler software Copyright © 2001-2003 James House. All rights reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product. This product uses and includes within its distribution, software developed by the Apache Software Foundation (http://www.apache.org/).

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

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 Sonic software, which includes software developed by Apache Software Foundation (http://www.apache.org/). Copyright © 1999-2000 The Apache Software Foundation. All rights reserved. The names “Ant”, “Axis”, “Xalan,” “FOP,” “The Jakarta Project”, “Tomcat”, “Xerces” and/or “Apache Software Foundation” must not be used to endorse or promote products derived from the Product without prior written permission. Any product derived from the Product may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission. For written permission, please contact apache@apache.org.

OpenEdge includes Sonic software, which includes software Copyright © 1999 CERN - European Organization for Nuclear Research. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

OpenEdge includes Sonic software, which includes software developed by ExoLab Project (http://www.exolab.org/). Copyright © 2000 Intalio Inc. All rights reserved. The names “Castor” and/or “ExoLab” must not be used to endorse or promote products derived from the Products without prior written permission. For written permission, please contact info@exolab.org. Exolab, Castor and Intalio are trademarks of Intalio Inc.

OpenEdge includes Sonic software, which includes software developed by IBM. Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. 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. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

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

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

OpenEdge includes Sonic software, which includes software developed by the University Corporation for Advanced Internet Development http://www.ucaid.edu Internet2 Project. Copyright © 2002 University Corporation for Advanced Internet Development, Inc. All rights reserved. Neither the name of OpenSAML nor the names of its contributors, nor Internet2, nor the University Corporation for Advanced Internet Development, Inc., nor UCAID may be used to endorse or promote products derived from this software and products derived from this software may not be called OpenSAML, Internet2, UCAID, or the University Corporation for Advanced Internet Development, nor may OpenSAML appear in their name without prior written permission of the University Corporation for Advanced Internet Development. For written permission, please contact opensaml@opensaml.org.

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 Vermont Views Terminal Handling Package software developed by Vermont Creative Software. Copyright © 1988-1991 by Vermont Creative Software.

OpenEdge includes XML Tools, which includes versions 8.9 of the Saxon XSLT and XQuery Processor from Saxonica Limited (http://www.saxonica.com/) which are available from SourceForge (http://sourceforge.net/projects/saxon/). The Original Code of Saxon comprises all those components which are not explicitly attributed to other parties. The Initial Developer of the Original Code is Michael Kay. Until February 2001 Michael Kay was an employee of International Computers Limited (now part of Fujitsu Limited), and original code developed during that time was released under this license by permission from International Computers Limited. From February 2001 until February 2004 Michael Kay was an employee of Software AG, and code developed during that time was released under this license by permission from Software AG, acting as a "Contributor". Subsequent code has been developed by Saxonica Limited, of which Michael Kay is a Director, again acting as a "Contributor". A small number of modules, or enhancements to modules, have been developed by other individuals (either written especially for Saxon, or incorporated into Saxon having initially been released as part of another open source product). Such contributions are acknowledged individually in comments attached to the relevant code modules. All Rights Reserved. The contents of the Saxon files are subject to the Mozilla Public License Version 1.0 (the "License"); you may not use these files except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/ and a copy of the license can also be found in the
installation directory, in the c:/OpenEdge/licenses folder. 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.

OpenEdge includes XML Tools, which includes Xs3P v1.1.3. The contents of this file are subject to the DSTC Public License (DPL) Version 1.1 (the "License"); you may not use this file except in compliance with the License. A copy of the license can be found in the installation directory, in the c:/OpenEdge/licenses folder. 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 xs3p. The Initial Developer of the Original Code is DSTC. Portions created by DSTC are Copyright © 2001, 2002 DSTC Pty Ltd. All rights reserved.

OpenEdge includes YAJL software Copyright 2007, Lloyd Hilaiel. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of Lloyd Hilaiel nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
What’s New in 10.2B

This chapter summarizes new and enhanced features in OpenEdge® Release 10.2B. For more detailed information about a feature, see the appropriate OpenEdge documentation referenced. This chapter contains the following sections:

- Installation
- OpenEdge startup parameters
- OpenEdge Architect
- OpenEdge GUI for .NET
- ABL enhancements
- Open Client enhancements
- AppServer enhancements
- OpenEdge interceptors for Actional
- Deployment
- OpenEdge RDBMS
- OpenEdge Replication
- OpenEdge Management and OpenEdge Explorer
- DataServers
Installation

Updates to the OpenEdge Release 10.2B installation include:

- OpenEdge Management and OpenEdge Explorer are integrated into the core product installation.

  New in Release 10.2B, installation of OpenEdge Management and OpenEdge Explorer is part of the core product installation. For OpenEdge Management, you must supply the appropriate serial number and control number. During the installation process you are prompted for the destination directories for the OpenEdge Management product and work directories (separate from the core OpenEdge product and work directories.) For more information see the “OpenEdge Management and OpenEdge Explorer” section on page 1–40.

- Infragistics online Help is automatically installed.

  The core installation automatically installs the online Help for the Infragistics controls when you install the OpenEdge Ultra Controls for .NET product.

- Installation of OpenEdge Architect plug-ins to multiple Eclipse frameworks.

  When installing OpenEdge Architect, you have the option to integrate the OpenEdge Architect plug-ins into additional Eclipse frameworks. Supported framework versions are 3.4.2 and 3.5.0.

For details about how to run an OpenEdge installation, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Getting Started: Installation and Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Installation online help</td>
</tr>
</tbody>
</table>
OpenEdge startup parameters

OpenEdge 10.2B provides new and modified startup parameters in support of various enhancements:

- Blocks in Alternate Buffer Pool (-B2)
- Nested Blocks (-nb)
- Prompt for Passphrase (-Passphrase)
- Re-usable Objects Cache (-reusableObjects)

### Blocks in Alternate Buffer Pool (-B2)

<table>
<thead>
<tr>
<th>Operating system and syntax</th>
<th>UNIX</th>
<th>Windows</th>
<th>-B2 ( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use with</td>
<td>Maximum value</td>
<td>Minimum value</td>
<td>Single-user default</td>
</tr>
<tr>
<td>CC, DBS</td>
<td>1,000,000,000 or 125,000,000(^1)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\(^1\) The maximum value of the sum of \(-B + -B2\), 1,000,000,000 for 64-bit platforms; 125,000,000 for 32-bit platforms.

The maximum is system dependent and limited by available memory.

\( n \)

The number of blocks allocated in the alternate buffer pool.

Use Blocks in Alternate Buffer Pool (-B2) to specify the number of blocks in the alternate buffer pool. The optimum value depends on your application.

You cannot specify Blocks in Alternate Buffer Pool for a Replication target database.

For more information about Alternate Buffer Pool, see:

**Manual:** *OpenEdge Data Management: Database Administration*
Nested Blocks (-nb)

Use Nested Blocks (-nb) to limit the maximum number of nested procedure blocks allowed.

<table>
<thead>
<tr>
<th>Operating system and syntax</th>
<th>UNIX</th>
<th>Windows</th>
<th>-nb n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use with</td>
<td>Maximum value</td>
<td>Minimum value</td>
<td>Single-user default</td>
</tr>
<tr>
<td>Client Session</td>
<td>20000</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

\( n \)

The maximum number of nested blocks.

By default, the maximum number of nested blocks is 100. Nested block entries are allocated in memory; each requires 32 bytes. Therefore, decrease -nb only if memory is severely limited.

The -nb limit is a soft limit. If your application needs to exceed the limit, the AVM issues a WARNING message and automatically increases the number of nested procedure blocks allowed. The WARNING is written to the current output (if there is current output) and to the log file (LG).

You can force the AVM to adhere to the specified -nb limit by starting the session with the Hardlimit (-hardlimit) startup parameter. When you use the -hardlimit startup parameter, the AVM issues the WARNING message when you exceed the Nested Blocks (-nb) limit. It also issues a message that a resource limit was reached and raises an untrappable STOP condition. Note that -hardlimit also enforces the limits set by the Directory Size (-D), the Local Buffer Size (-l), and the Maximum Memory (-mmax) startup parameters, so the STOP condition is raised when any of the specified limits is reached.

For more information about nested blocks, see:

Manuals:  
OpenEdge Deployment: Managing ABL Applications  
OpenEdge Getting Started: ABL Essentials
Prompt for Passphrase (-Passphrase)

Use Prompt for Passphrase to open an encryption-enabled database that is configured for manual start, or to override the autostart passphrase.

<table>
<thead>
<tr>
<th>Operating system and syntax</th>
<th>UNIX</th>
<th>Windows</th>
<th>-Passphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use with</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Database utilities and servers, and non-networked self-service clients indicate that they want to be prompted for key store authentication by adding Prompt for Passphrase (-Passphrase) to the command line. Prompt for Passphrase (-Passphrase) prompts for the key store passphrase. If you cannot successfully authenticate the key store, you can not open the database.

For more information about Key Store Passphrase, and Transparent Data Encryption in general, see:

**Manuals:**
- *OpenEdge Getting Started: Core Business Services*
- *OpenEdge Data Management: Database Administration*

Re-usable Objects Cache (-reusableObjects)

Use Re-usable Objects Cache (-reusableObjects) to specify the number of deleted class objects that the AVM stores for later re-initialization.

<table>
<thead>
<tr>
<th>Operating system and syntax</th>
<th>UNIX</th>
<th>Windows</th>
<th>-reusableObjects cache-size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use with</td>
<td>2 GB</td>
<td>0</td>
<td>25</td>
</tr>
</tbody>
</table>

`cache-size`

Number of deleted class objects to store for later re-initialization.

This parameter can improve the performance of applications that use ABL classes. When a class object is deleted, either by the DELETE OBJECT statement or through garbage collection, the AVM caches the object. When the AVM instantiates the same class later, the stored object is re-initialized and removed from the cache. This re-use saves much of the overhead of instantiating a class.
Note:  The cache only stores class-based objects. However, it does not store .NET classes, .NET-derived ABL classes, classes with STATIC elements, or session-compiled classes.

The cached class objects continue to hold some resources. You might want to modify the settings of Directory Size (-d) and Maximum Memory (-mmax) to account for the number of cached objects.

For more information about the re-use of objects, see the DELETE OBJECT statement in OpenEdge Development: ABL Reference and the object life-cycle section in OpenEdge Development: Object-oriented Programming.
OpenEdge Architect

The 10.2B Release of the OpenEdge Architect includes new and revised features in the following areas:

- Upgrade to Eclipse 3.4.2
- File layout changes
- New video demonstrations
- Upgrade of OpenEdge Ultra Controls for .NET
- Update Assembly References Tool
- Support for ABL language improvements
- ABL Editor context-filtered code assistance
- Improved ProBindingSource Designer
- Global Toolbox option
- Shared AVM
- Performance improvements
- AppServer support
- Class Browser enhancements
- Support for AppBuilder shortcut keys

Upgrade to Eclipse 3.4.2

The OpenEdge Architect 10.2B release includes Eclipse 3.4.2. The 10.2A release of OpenEdge Architect shipped with Eclipse 3.4. The upgrade to Eclipse 3.4.2 includes a new Welcome page that provides the following:

- Links to various sources of information for tooling
- Easy, up-to-date access to additional “Getting started with OpenEdge Architect” material, including short videos and white papers developed by OpenEdge Knowledge Services
- Better integration with the welcome pages of other Eclipse-based Progress products and Eclipsed-based third-party plug-ins

By leveraging the Eclipse universal intro page architecture, the new OpenEdge Architect Welcome page provides a better user experience when more than one Progress product is installed in a single Eclipse environment. For example, users of both OpenEdge Architect and Progress® Sonic® Workbench will get Welcome page content from both products in one place, creating efficiencies in a single view. Similarly, the Welcome pages from third party plug-ins that also leverage the Eclipse universal intro page architecture will also be better integrated.

In addition, OpenEdge Architect 10.2B is certified to run on Eclipse 3.5.
File layout changes

To more fully support the Eclipse Equinox/p2 provisioning system for managing updates and installing new features, the file layout of OpenEdge Architect has been changed in Release 10.2B. This change involves the addition of several new directories that act as repositories to provision OpenEdge Architect and third-party plug-ins. Figure 1–1 shows the new directory structure of OpenEdge-install-dir/oeide.

![Figure 1–1: OpenEdge Architect file layout](image)

Note that the ../oeide/architect folder, which contained the installed OpenEdge Architect plug-ins in prior releases, no longer exists. OpenEdge Architect and all other installed plug-ins are now installed in ../oeide/eclipse/plugins.

Also note that, in addition to the Architect_repo folder for OpenEdge Architect, there are two repository folders to provision third-party plug-ins. One is for third-party tools that have Eclipse 3.4.2 dependencies and the other is for third-party tools that have Eclipse 3.5.0 dependencies. OpenEdge Architect 10.2B supports both of these Eclipse platforms.

New video demonstrations

There are a variety of video demonstrations, hosted on the Progress Communities Web site, that illustrate the features of OpenEdge Architect.

For demos describing how to get started with OpenEdge Architect, go to [http://communities.progress.com/pcom/docs/DOC-101140](http://communities.progress.com/pcom/docs/DOC-101140).

For demos that show a variety of OpenEdge Architect features, go to [http://communities.progress.com/pcom/docs/DOC-101504](http://communities.progress.com/pcom/docs/DOC-101504).

For direct links to appropriate demos from various topics, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Getting Started: Introducing the OpenEdge Architect Visual Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>OpenEdge Architect</td>
</tr>
</tbody>
</table>
Upgrade of OpenEdge Ultra Controls for .NET

Customers who license OpenEdge Ultra Controls for .NET receive version 9.2 of the controls with OpenEdge 10.2B. In that respect, OpenEdge 10.2B differs from OpenEdge 10.2A, which includes OpenEdge Ultra Controls version 8.1.

Any OpenEdge 10.2A r-code that you created that contains UltraControls for .NET controls needs to be recompiled for 10.2B. The names of the Infragistics controls in version 8.1 are not the same as the names of the controls in version 9.2.

Update Assembly References Tool

OpenEdge 10.2B includes a new Update Assembly References tool that helps you to synchronize assembly references in existing projects with new versions of those assemblies. Each time a new version of a control set is installed, it is necessary to update these references. For example, the change from version 8.1 to version 9.2 of the OpenEdge Ultra Controls necessitates such an update.

You can invoke the Update Assembly References tool either from the OpenEdge→Migration menu or from a command prompt.

For more information about the Update Assembly References tool, see:

| Online help: | Visual Designer |

Support for ABL language improvements

OpenEdge 10.2B introduces ABL support for:

- Abstract classes and data members
- Strongly typed events
- Accessing .NET generics (definition of new generics in ABL is not supported)

In OpenEdge Architect, code-generation wizards, code assistance, the Outline view, Quick Outline, and the Debugger have been extended to support these language improvements.

For more information about support for ABL improvements in OpenEdge Architect, see:

| Online help: | ABL Editor, Debugger, Visual Designer |

For more information about other ABL enhancements in this release, see the “ABL enhancements” section on page 1–14.
ABL Editor context-filtered code assistance

By default, code-completion assistance now initially offers only those proposals that are relevant in the context of the current source file. You can press **CTRL+SPACE** to toggle the list between context-filtered suggestions and all suggestions. A preference setting is available to disable filtering of assistance proposals according to context.

For more information about ABL Editor context-filtered code assistance, see:

| Online help: | ABL Editor |

Improved ProBindingSource Designer

In the Visual Designer, usability improvements have been made to the tool that helps you to define the schema displayed by a data-bound control. Enhancements to the ProBindingSource Designer include support for drag-and-drop operations, streamlined support for importing a schema from a file or from a database, and an updated user interface.

For more information about the improved ProBindingSource Designer, see:

| Online help: | Visual Designer |

Global Toolbox option

The Visual Designer now offers the option of sharing a common Toolbox among multiple projects. The option to use a dedicated Toolbox for a project remains available. A new project property setting controls this behavior for each project.

For more information about the Global Toolbox option, see:

| Online help: | Visual Designer |

Shared AVM

The ABL Virtual Machine (AVM) is a platform for compiling and running ABL code. In prior releases of OpenEdge Architect, each project in a workspace had its own running AVM instance.

Release 10.2B of OpenEdge Architect introduces a shared AVM, which is a single AVM that can be used by any or all projects in a workspace. The shared AVM is useful if multiple projects require the same database connection, PROPATH, AppBuilder, Assemblies, and AVM startup properties. The shared AVM is advantageous because it uses fewer memory and CPU resources.
To configure a shared AVM:

1. Select **Window → Preferences** from the main menu bar.
2. Select **OpenEdge Architect → Shared AVM** in the tree view of the Preferences dialog.
3. Set the AVM startup parameters on the Shared OpenEdge AVM preference page.
4. Set the other properties in the child nodes that affect all the projects that use the shared AVM.

In the New OpenEdge Project wizard (**File → New → OpenEdge Project**), you can choose whether you want to use an individual, project-specific AVM or a shared AVM. For existing projects in the workspace, you can go to the OpenEdge property page for the project and select the **Use shared AVM** option.

For more information about the shared AVM, see:

**Online help:** Getting Started, OpenEdge Projects

### Performance improvements

The performance of OpenEdge Architect has been improved in the following areas:

- Updating the Outline view when changes occur
- Expanding include files
- Opening files
- Responding to user input
- Adding code sections (functions, methods, etc.) using the various source-editing wizards

### AppServer support

Release 10.2B supports the OpenEdge AppServer in OpenEdge Architect, allowing you to do the following:

- Manage AppServers from within OpenEdge Architect by defining connections to AppServer brokers.
- Use one tool—OpenEdge Architect—to perform multiple AppServer functions, such as starting or stopping servers or adding or trimming AppServer agents.

For additional AppServer-related tasks, use Progress Explorer, OpenEdge Management (with the required license), or OpenEdge Explorer. You can access OpenEdge Explorer and/or OpenEdge Management from within OpenEdge Architect.
Create and publish an application to an AppServer, which allows you to test the application in an environment outside of the OpenEdge Architect workspace.

Use the Eclipse launch configuration framework to set certain AppServer properties and create and store alternate configurations for each AppServer in your environment.

Monitor AppServer broker and server status details through a server monitor view.

For more information about the AppServer in OpenEdge Architect, see:

| Online help: | AppServer |

### Class Browser enhancements

The Class Browser includes the following enhancements in Release 10.2B:

- The description of a class, method, property, or event now includes `ABSTRACT`, if applicable.
- The Class Browser view now displays .NET generic types.
- The Class Browser view recognizes the ABL events declared in an ABL class and populates them in the events section of class.
- The ABL Syntax section for an event, as viewed in the Summary pane, now provides sample code that you can use to define a method handler.
- The Class Browser uses the `PROPATH` specified at the project level when processing project classes. For a project using the shared AVM, the Class Browser shows the assemblies and the `PROPATH` defined for the shared AVM.

For more information about the Class Browser, see:

| Online help: | Class Browser |

### Support for AppBuilder shortcut keys

The Keys preference page (`Window → Preferences → General → Keys`) has a new choice for key bindings. You can select `AppBuilder` in the `Scheme` box to implement most of the AppBuilder shortcut keys.

For more information about AppBuilder shortcut keys, see:

| Online help: | AppBuilder and Other OpenEdge Tools |
OpenEdge GUI for .NET

OpenEdge Release 10.2B provides support for the following features:

- New ABL support for .NET features
- New ProBindingSource properties

New ABL support for .NET features

Release 10.2B provides enhanced ABL support for the following features:

- **.NET abstract classes** — ABL supports the inheritance and implementation of additional .NET abstract class members. For more information, see the “Abstract classes” section on page 1–15.

- **.NET events** — ABL adds events as members of ABL classes that function consistently with previous support for .NET events. This allows ABL applications to more naturally incorporate .NET events into an ABL-extended .NET class hierarchy. For more information, see the “Class events” section on page 1–16.

- **.NET generic types** — ABL supports instantiation and references to .NET generic types. For more information, see the “.NET generic types” section on page 1–17.

- **.NET stack traces** — ABL supports access to .NET stack traces in the Debug Alert Help dialog box. For more information, see the “.NET stack trace for Debug Alert” section on page 1–21.

New ProBindingSource properties

Two new properties added to the ProBindingSource include:

- **MaxDataGuess** — An estimated number of records for the ProBindingSource query’s result set. This property can improve performance visualizing .NET controls with large result sets.

- **NoLOBs** — Specifies whether or not the AVM ignores BLOB or CLOB fields while executing the ProBindingSource’s Assign( ) method or the CURRENT-CHANGED function.

For more information about accessing .NET objects and using the OpenEdge GUI for .NET, see:

| Manuals: | OpenEdge Development: GUI for .NET Programming  
|          | OpenEdge Development: GUI for .NET Mapping Reference  
|          | OpenEdge Development: ABL Reference |
| Online help: | ABL Reference online help |
ABL enhancements

ABL includes the following enhancements in this release:

- Re-use of ABL class objects
- Abstract classes
- Class events
- .NET generic types
- ABL Compiler XREF output
- ABL logging
- Enhanced reflection, dynamic instantiation, and invocation in object-oriented ABL
- Dynamic invocation of DLLs and UNIX shared libraries
- Class member namespace consistency
- .NET stack trace for Debug Alert
- Temp-table and ProDataSet improvements
- Converting temp-tables and ProDataSets to and from JSON
- STOP-AFTER phrase provides timed STOP condition for DO, FOR, and REPEAT blocks
- CANCEL-REQUESTS-AFTER method provides timed STOP condition for remote procedure calls
- Connecting to an encryption-enabled database

For more information about new and modified language elements referenced in this section, see:

<table>
<thead>
<tr>
<th>Manual</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>ABL Reference online help (includes both reference sources)</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Architect online help</td>
</tr>
</tbody>
</table>

Re-use of ABL class objects

When the AVM deletes an instance of an ABL class, it now places the object data in a re-usable objects cache. Whenever the AVM instantiates an object of an ABL class, it first looks to see if there is a re-usable instance of the object in that cache. This re-use improves the performance of ABL object instantiation.

For more information, see the “Re-usable Objects Cache (-reusableObjects)” section on page 1–5 and the DELETE OBJECT statement in OpenEdge Development: ABL Reference.
Abstract classes

ABL supports the definition of abstract classes and class members, including methods, properties, and class events (see the “Class events” section on page 1–16). Abstract classes provide a means to define class hierarchies that support different implementations for a common set of abstract members, as well as to share common implementations for a set of non-abstract members. An ABL class creates its own implementation of an abstract class member by inheriting and overriding the abstract member defined in an abstract super class.

ABL abstract classes also include enhanced ABL support for .NET abstract classes. Essentially, with few restrictions, an ABL class can inherit and implement any .NET abstract class member similar to how it inherits and implements an equivalent ABL abstract class member, including .NET data type and signature mapping, as required.

ABL supports abstract classes with the following updates:

- **Class-based method call** — Updated to call an abstract method
- **Class-based property access** — Updated to access an abstract property
- **CLASS statement** — Updated with a new ABSTRACT option
- **COMPILE statement** — Updated with additional identifiers in the output for the XREF and XREF-XML options (see the “ABL Compiler XREF output” section on page 1–18)
- **DEFINE EVENT statement** — New statement for defining class events that includes the ABSTRACT and OVERRIDE options, and general syntax for defining abstract event prototypes
- **DEFINE PROPERTY statement** — Updated to include new ABSTRACT and OVERRIDE options, and general syntax for defining abstract property prototypes
- **DYNAMIC-NEW statement** — Updated to prevent instantiation of abstract classes
- **IsAbstract( ) method on the Progress.Lang.Class class** — New method to indicate if a class is abstract
- **METHOD statement** — Updated to include a new ABSTRACT option, additional functionality for the OVERRIDE option, and general syntax for defining abstract method prototypes
- **NEW function (classes) and statement** — Updated to prevent instantiation of abstract classes
- **Publish( ) event method** — New method to publish an ABL class event, including support to publish an ABL class event that implements an inherited .NET or ABL abstract event
- **Subscribe( ) event method** — Updated and enhanced for subscribing event handlers to both ABL and .NET class events, including abstract events
- **Unsubscribe( ) event method** — Updated and enhanced for cancelling event handler subscriptions to both ABL and .NET class events, including abstract events
For more information about ABL support for abstract classes, see:

### Manuals:

- *OpenEdge Development: ABL Reference*
- *OpenEdge Development: GUI for .NET Programming*
- *OpenEdge Development: Object-oriented Programming*

## Class events

ABL supports the definition of events as members of an ABL class. This support also incorporates and enhances the previous support for .NET events in ABL. Like .NET events for .NET classes, ABL class events allow an ABL class to notify an ABL application of a given condition by publishing an associated event when the condition is identified. Publishing the event then executes one or more methods or internal procedures that are registered as handlers for the event. Each such method or internal procedure must have a signature that is compatible with the method signature defined for the event.

The enhanced .NET support includes the ability to define ABL class event signatures with reference to .NET event delegates. This allows an ABL class to override and implement an inherited .NET abstract event or to implement a .NET interface event. You can also define an ABL class event with reference to a .NET delegate for any other application purpose.

ABL supports class events using a model similar to ABL named events, which allow you to publish events and to dynamically subscribe or unsubscribe event handlers. However, ABL supports class events using a separate mechanism that respects the strong typing of events as members of an ABL class.

ABL supports class events with the following updates:

- **CLASS statement** — Updated to support events as members of a class
- **COMPILE statement** — Updated with additional identifiers in the output for the XREF and XREF-XML options (see the “ABL Compiler XREF output” section on page 1–18)
- **DEFINE EVENT statement** — New statement for defining events as members of a class or interface
- **INTERFACE statement** — Updated to support class events as members of an interface
- **Publish( ) event method** — New method for publishing ABL class events
- **Subscribe( ) event method** — Updated and enhanced for subscribing event handlers to both ABL and .NET class events
- **Unsubscribe( ) event method** — Updated and enhanced for cancelling event handler subscriptions to both ABL and .NET class events

Invoking the `Publish( )`, `Subscribe( )`, or `Unsubscribe( )` method also generates a log message when you enable logging for the 4GLTrace log entry type. For more information, see the “ABL logging” section on page 1–18.
For more information about ABL support for class events, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: GUI for .NET Mapping Reference</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
</tbody>
</table>

.NET generic types

ABL allows you to instantiate a .NET generic class or otherwise reference a .NET generic type. A generic type (sometimes called a parameterized type) is a class or interface type whose definition supports multiple implementations, each of which operates on a different set of data types that are specified as parameters to the generic type name. Thus, the type name you use for a generic type results in a customized version of the implementation that uses the type parameters specified in the name.

.NET supports two forms of type names for a generic type:

- An open type name, which specifies the generic type definition
- A constructed type name, which provides the type parameters and thus results in a customized implementation at runtime

ABL only supports references to constructed .NET generic type names. Thus, you cannot inherit from a .NET generic class, implement a .NET generic interface, or otherwise define or reference an open generic type in ABL. In this release, you also cannot call a .NET generic method in ABL. You can only instantiate existing generic classes and or reference static generic class types in order to access non-generic public members of those class instances and types.

ABL supports .NET generic types with the following updates:

- **CAST functions** — Updated to support casts involving .NET generic types
- **Data types** — Updated with support for .NET generic types
- **ToString( ) method** — Updated to return a string that represents the constructed type
- **Type-name syntax** — Updated to reference constructed .NET generic types
- **USING statement** — Updated to support unqualified references to .NET generic type names

For more information about ABL support for .NET generic types, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
</tbody>
</table>
ABL Compiler XREF output

The output for the XREF option of the COMPILE statement has updated reference types, as follows:

- **CLASS** — Updated with a new ABSTRACT element in the object identifier
- **EVENT** — New reference type
- **METHOD** — Updated with a new ABSTRACT element in the object identifier
- **PUBLISH** — Updated with an optional class type name qualifying the event-name element in the object identifier
- **SUBSCRIBE** — Updated object identifier with an optional class type name qualifying the event-name element, a new handler-name element (optionally class-qualified), and an optional list of parameter elements
- **UNSUBSCRIBE** — Updated object identifier with an optional class type name qualifying the event-name element, a new handler-name element (optionally class-qualified), and an optional list of parameter elements

For more information about the updated XREF option output on the COMPILE statement, see:

| Manuals: | OpenEdge Development: ABL Reference |

ABL logging

ABL provides enhanced support for the following entry message types controlled by the 4GLTrace log entry type:

- **Publish** — Updated entry message type that is also logged when you call the new Publish( ) event method either on a native ABL class event or on an ABL class event that implements an inherited .NET abstract event.
- **Subscribe** — Updated entry message type that is also logged when you call the existing Subscribe( ) event method to subscribe an event handler for an ABL or .NET class event.
- **Unsubscribe** — New entry message type that is logged when you call the existing Unsubscribe( ) event method to unsubscribe an event handler for an ABL or .NET class event.

For more information about support for ABL class events, see the “Class events” section on page 1–16.

For more information about the 4GLTrace log entry type, see:

| Manuals: | OpenEdge Development: Debugging and Troubleshooting |
Enhanced reflection, dynamic instantiation, and invocation in object-oriented ABL

To make it easier for you to incorporate object-oriented ABL in your application, the following ABL elements were modified or added in support of reflection, dynamic instantiation, and invocation:

- **DYNAMIC-NEW function** — Updated to support full invocation of overload constructors. An application can now instantiate a class that has overloaded constructors, where the constructors differ by number of parameters, parameter type, or mode.

- **DYNAMIC-INVoke function** — New function allows for the dynamic invocation of a method where the name is not known at compile time, but whose parameters are known and provided at compile time.

- **HasStatics( ) method on the Progress.Lang.Class class** — New method returns TRUE if a given class has any static members.

- **HasWidgetPool( ) method on the Progress.Lang.Class class** — New method returns TRUE if a given class has a widget pool.

- **Invoke( ) method on the Progress.Lang.Class class** — New method that invokes a class-based method when the method name and any parameters are only known at run time. To support this method, the Progress.Lang.ParameterList class was added to build a list of parameter values dynamically.

- **IsA( ) method on the Progress.Lang.Class class** — New method returns TRUE if the specified class type name expression or Progress.Lang.Class object is within the class hierarchy of the Progress.Lang.Class type or implements an interface identified by the Progress.Lang.Class type.

- **IsAbstract( ) method on the Progress.Lang.Class class** — New method returns TRUE if a given class is abstract.

- **New( ) method on the Progress.Lang.Class class** — New method that dynamically creates an instance of a class when the class name and any parameters passed to it are only known at run time. To support this method the Progress.Lang.ParameterList class was added to build a list of parameter values dynamically.

- **Progress.Lang.ParameterList class** — Stores the parameter values for a specific method or constructor call. The Progress.Lang.ParameterList class contains these public members:
  - **Clear( ) method** — Releases all information in the Progress.Lang.ParameterList object and sets the number of parameters to zero
  - **NumParameters property** — Identifies the number of parameters in a Progress.Lang.ParameterList object
  - **SetParameter( ) method** — Populates elements in the Progress.Lang.ParameterList object
For more information about reflection, dynamic instantiation, and invocation support in object-oriented programming with ABL, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td>Online help:</td>
<td>ABL Reference online help</td>
</tr>
</tbody>
</table>

**Dynamic invocation of DLLs and UNIX shared libraries**

The call object handle has been extended to support the dynamic invocation of Windows Dynamic Link Library (DLL) routines and UNIX shared library routines, hereafter referred to as *shared libraries*.

The following ABL elements are either modified or added in support of shared library routines:

- **CALL–NAME attribute (Call object handle)** — Updated to indicate the name of a shared library routine you want to invoke dynamically
- **CALL–TYPE attribute (Call object handle)** — Updated with the DLL–CALL–TYPE keyword constant to indicate your application is invoking a shared library routine
- **LIBRARY attribute (Call object handle)** — New attribute that specifies the name of a shared library
- **LIBRARY–CALLING–CONVENTION attribute (Call object handle)** — New attribute that specifies the shared library calling convention
- **ORDINAL attribute (Call object handle)** — New attribute that specifies the number of the Windows DLL entry point to invoke
- **PERSISTENT attribute (Call object handle)** — Modified to indicate if the shared library is persistent
- **RETURN–VALUE (Call object handle)** — Modified to support shared library return values
- **RETURN–VALUE–DLL–TYPE attribute (Call object handle)** — New attribute that specifies the expected data type returned from a Windows DLL routine
- **SET–PARAMETER method (Call object handle)** — Modified to specify parameter values to be passed to a shared library routine

For more information about dynamic invocation of shared libraries, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>Web paper:</td>
<td>Dynamic Call Object</td>
</tr>
</tbody>
</table>
Class member namespace consistency

Object-oriented programming support in ABL now defines six distinct namespaces for naming class members:

- Methods
- Variables, properties, and events
- Buffers and temp-tables
- ProDataSets
- Queries
- Data-Sources

In addition, there are two additional implicit members of a class that participate in namespaces: database tables and table (buffer) fields.

Each class member must have a unique name within its namespace and within its class hierarchy, depending on the access mode, overriding, and overloading of the member.

For more information about class member namespace consistency, see:

Manuals:  
OpenEdge Development: ABL Reference  
OpenEdge Development: Object-oriented Programming

.NET stack trace for Debug Alert

.NET Exceptions can be thrown when working with .NET objects in ABL. When you do not handle these Exceptions—there is no CATCH or NO-ERROR logic present—and you specify the Debug Alert (-debugalert) startup parameter or you set SESSION:DEBUG-ALERT to TRUE, the AVM adds the .NET stack trace to the Debug Alert information after the ABL call stack. The .NET stack trace is added both in the Debug Alert Help dialog box and in the client log (when -clientlog is specified). The top of the stack (most recent call) is displayed at the top of the trace listing.

For more information about .NET stack trace for Debug Alert, see:

Manuals:  
OpenEdge Development: ABL Reference  
OpenEdge Development: GUI for .NET Programming
Temp-table and ProDataSet improvements

Several improvements were made for working with temp-tables and ProDataSets, including:

- The requirement that source and target temp-tables be named differently was removed for these handle-based methods:
  - `COPY-DATASET( )` method (ProDataSet object handle), in REPLACE mode
  - `COPY-TEMP-TABLE( )` method (Temp-table object handle), in REPLACE mode
  - `GET-CHANGES( )` method (ProDataSet object handle), in `get-parent` mode
  - `MERGE-CHANGES( )` method (Buffer object handle, ProDataSet object handle)
  - `MERGE-ROW-CHANGES( )` method (Buffer object handle)
- The `NAME` attribute of the Temp-table object handle is writable for dynamic and AVM-generated temp-tables. This feature affects how you use the following ABL elements:
  - `FILL-WHERE-STRING` attribute (Data-source object handle)
  - `NAME` attribute (Temp-table object handle)
  - `QUERY-PREPARE( )` method (Object query handle)
  - `TEMP-TABLE-PREPARE( )` method (Temp-table object handle)
  - `WHERE-STRING` attribute (Data-relation object handle)
- When filling a ProDataSet and `FILL-MODE` is REPLACE, the BEFORE-ROW-FILL event handler is run after the original ProDataSet row to be replaced is copied into the ProDataSet buffer.
- The `ORIGIN-HANDLE` attribute is copied from the source temp-table to the target temp-table in all cases when using the `COPY-DATASET( )` method and the `COPY-TEMP-TABLE( )` method. Prior to 10.2B, `ORIGIN-HANDLE` was only copied from the source to the target if the source had before-table records.

For more information about these temp-table and ProDataSet improvements, see:

| Manual:                      | OpenEdge Development: ABL Reference |

**Converting temp-tables and ProDataSets to and from JSON**

OpenEdge Release 10.2B includes support for converting temp-tables and ProDataSets to and from JavaScript Object Notation (JSON) data. This feature parallels the existing functionality for converting these objects to and from XML data. This feature includes some generalized behavior for serializing data that affects the existing XML functionality.
Table 1–1 lists the new ABL elements that support JSON.

Table 1–1: New ABL elements to support JSON

<table>
<thead>
<tr>
<th>ABL element</th>
<th>Applies to . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERIALIZE-HIDDEN option</td>
<td>Field-options phrase for DEFINE TEMP-TABLE</td>
</tr>
<tr>
<td>SERIALIZE-NAME option</td>
<td>DEFINE DATASET statement</td>
</tr>
<tr>
<td></td>
<td>DEFINE TEMP-TABLE statement</td>
</tr>
<tr>
<td></td>
<td>Field-options phrase for DEFINE TEMP-TABLE</td>
</tr>
<tr>
<td>IS-JSON attribute</td>
<td>WEB-CONTEXT system handle</td>
</tr>
<tr>
<td>SERIALIZE-HIDDEN attribute</td>
<td>Buffer-field object handle</td>
</tr>
<tr>
<td>SERIALIZE-NAME attribute</td>
<td>Buffer object handle</td>
</tr>
<tr>
<td></td>
<td>Buffer-field object handle</td>
</tr>
<tr>
<td></td>
<td>ProDataSet object handle</td>
</tr>
<tr>
<td></td>
<td>Temp-table object handle</td>
</tr>
<tr>
<td>READ-JSON() method</td>
<td>Buffer object handle</td>
</tr>
<tr>
<td></td>
<td>ProDataSet object handle</td>
</tr>
<tr>
<td></td>
<td>Temp-table object handle</td>
</tr>
<tr>
<td>WRITE-JSON() method</td>
<td>Buffer object handle</td>
</tr>
<tr>
<td></td>
<td>ProDataSet object handle</td>
</tr>
<tr>
<td></td>
<td>Temp-table object handle</td>
</tr>
</tbody>
</table>

To generalize the behavior for serializing data, the following pairs of options/attributes interact:

- SERIALIZE-HIDDEN and XML-NODE-TYPE
- SERIALIZE-NAME and XML-NODE-NAME

For more information about ABL support for JSON, see:

**Manuals:**
- OpenEdge Development: ABL Reference
- OpenEdge Development: Working with JSON

STOP-AFTER phrase provides timed STOP condition for DO, FOR, and REPEAT blocks

Many applications need the ability to terminate operations if the operations take more time than the application design specifies. This is especially true if the application appears unresponsive to you while waiting for a long-running operation to complete.

The ABL now provides the STOP-AFTER phrase on the DO, FOR, and REPEAT block statements. The phrase allows you to specify an integer expression that evaluates to a number of seconds, referred to as the timer. When the timer runs out, a timeout occurs, causing the AVM to raise the STOP condition. Normal STOP handling then occurs, and you can alter the default handling, as usual, using the ON STOP phrase.
There are many potential use cases for this functionality, but the primary use cases are for canceling potentially long-running operations like:

- Dynamic queries
- Remote procedure calls

**STOP-AFTER phrase**

STOP-AFTER is an optional phrase that you can place in any order with other optional phrases in the DO, FOR, or REPEAT block statements. The syntax for the STOP-AFTER phrase is as follows:

**Syntax**

```
STOP-AFTER expression
```

STOP-AFTER specifies an integer timeout value. The integer expression specifies the number of seconds a block has until a timeout occurs. If it is an iterating block, then the timer specifies the timer for each iteration of the block. If a timeout occurs, the AVM raises the STOP condition and default STOP condition handling occurs. Use an ON STOP phrase on the block (or an enclosing block) to alter the default STOP condition handling.

If the block iteration completes before the specified time expires, the timer resets to expression. In other words, the timer is limited to the scope of the block. If a block with a STOP-AFTER phrase encloses another block or calls another block without a STOP-AFTER phrase, the timer continues.

If a block with a STOP-AFTER phrase contains a nested block with a STOP-AFTER phrase, then each timer is in effect. If the outer block timer expires within the inner block, the STOP condition is raised even if the timer for the inner block has not expired.

For information about related features see the “CANCEL-REQUESTS-AFTER method provides timed STOP condition for remote procedure calls” section on page 1–24, and the “Configuring a default time limit for server applications” section on page 1–33.

For more information about the STOP-AFTER phrase in the DO, FOR, or REPEAT statement entries, see:

**Manual:**  
OpenEdge Development: ABL Reference

**CANCEL-REQUESTS-AFTER method provides timed STOP condition for remote procedure calls**

The CANCEL-REQUESTS-AFTER( ) method calls the CANCEL-REQUESTS( ) method on the server object after the specified number of seconds have elapsed. When called, all requests currently running or queued to run on the server object are canceled, regardless of when they were started.

If the method is called on a server object that already has a timer running, the timer is restarted with the new time interval, measured from that moment.
If the method is called with a parameter whose value is less than or equal to zero, then the timer on that server object is stopped.

**Return type:** LOGICAL  
**Applies to:** Server object handle

**Syntax**

```plaintext
CANCEL-REQUESTS(seconds)
```

*seconds*

This required integer parameter specifies the number of seconds before the `CANCEL-REQUESTS( )` method is called on the server object.

The method returns TRUE unless an error occurs while setting the timer. In this case, the method returns FALSE, and the time-out is not set.

Also see the “STOP-AFTER phrase provides timed STOP condition for DO, FOR, and REPEAT blocks” section on page 1–23, and the “Configuring a default time limit for server applications” section on page 1–33, and for additional information, see:

**Manuals:**  
*OpenEdge Development: ABL Reference*  
*OpenEdge Application Server: Developing AppServer Applications*

### Connecting to an encryption-enabled database

Connecting to a database enabled for Transparent Data Encryption may require modification to the CONNECT statement. If the database is not configured for autostart, you must supply the key store passphrase, as shown:

```plaintext
CONNECT mydb -1 -KeyStorePassPhrase "passphrase-here".
```

Your application must prompt you for the passphrase.

Transparent Data Encryption is available on all supported platforms except Unixware.

For more information on which products are available and the supported platforms, see

**Manuals:**  
*OpenEdge Getting Started: Installation and Configuration*

For more information about connecting to an encryption-enabled database, and Transparent Data Encryption in general, see

**Manuals:**  
*OpenEdge Data Management: Database Administration*  
*OpenEdge Development: Basic Database Tools*  
*OpenEdge Getting Started: Core Business Services*  
*OpenEdge Getting Started: Database Essentials*
Open Client enhancements

OpenEdge Release 10.2B includes the following enhancement related to Open Client:

- .NET Open Client supports strongly-named, unsigned assemblies
- New NoSchemaMarshal property for .NET Open Client

.NET Open Client supports strongly-named, unsigned assemblies

The Open Client for .NET now supports assemblies using strongly named but unsigned DLLs.

In previous releases, the .NET Open Client supported Digitally Signed assemblies as well as Digitally Signed and Strongly Named assemblies. With Release 10.2B, .NET Open Client supports Digitally Unsigned and Strongly Named assemblies.

**Note:** Support for the new assembly type is also available for OpenEdge version 10.1C and OpenEdge version 10.2A with the latest patches.

The digital signature process identifies details about the author of a particular assembly. It is generally considered standard practice to digitally sign an assembly. By default, .NET Open Client assemblies are digitally signed.

The strong naming process allows applications to precisely match the version of a DLL needed by a caller. In environments where several versions of an assembly may be needed to support multiple products or versions of the same product, strong naming is vital.

Because strong naming uses several criteria to precisely identify the version of a DLL, its use makes unauthorized code changes very difficult to implement. Thus, strong naming can also be seen as a security measure.

Strong naming entails the overhead of more careful code management. For this reason, you should carefully research the pros and cons of strong naming before enabling this option.

You can choose to use strongly named assemblies with or without digital signing. Strong naming with digital signing is the most common choice. Assemblies with strong naming but no digital signing are a less common use case, but may be required in some environments. You can now choose these assembly types in ProxyGen.

Changes to ProxyGen

On the .NET tab, the Use Strongnamed Runtime checkbox has been replaced with a new Runtime combo box. The combo box specifies which type of Open Client runtime assemblies (DLLs) you want to use for your Open Client project:

- **Digitally Signed** — Indicates that the DLLs will be digitally signed to identify the author, but will not be strongly named for precise DLL version matching. This is the default setting.

- **Strongnamed Signed** — Indicates that the DLLs will be both digitally signed to identify the author and strongly named for precise DLL version matching.

- **Strongnamed** — Indicates that the DLLs will not be digitally signed to identify the author and will be strongly named for precise DLL version matching.
The **Delay Sign** checkbox now generates an error message if you selected it when the **Runtime** combo box is set to **Digitally Signed**.

**New files**

To support the new assembly option, there are now three sets of .NET Open Client DLLs (Progress.ssl.dll, Progress.o4glrt.dll, and Progress.Messages.dll). They can be found in the following directories:

- InstallDir/dotnet/deploy/signed
- InstallDir/dotnet/deploy/strongnamed-signed
- InstallDir/dotnet/deploy/strongnamed

**New NoSchemaMarshal property for .NET Open Client**

PROGRESS.Session.NoSchemaMarshal, when set to **true**, prevents .NET from sending schema information to the AppServer for input-output and output DataSet and DataTable properties. The property applies to all DataSet and DataTable parameters for subsequent method calls.

This property does not effect marshalling from the AppServer to the .NET Open Client.
PROGRESS.Session.NoSchemaMarshal

**Data Type:** Boolean

**Session model:** Session-free and session-managed

When set to true, this property prevents .NET from sending schema information to the AppServer for input-output and output DataSet and DataTable properties. The property applies to all DataSet and DataTable parameters for subsequent method calls. Set the property before a proxy method is called. Reset the NoSchemaMarshal property back to false for subsequent proxy method calls if needed.

This property must be used for output only DataSet or DataTable parameters, and not the ProDataTable.SetNoSchemaMarshal( ) method.

This is because the c# compiler considers an output parameter unassigned and it must be assigned before its value is used, therefore, the .NET proxy code is not able to look at the output DataSet or DataTable object to detect if SetNoSchemaMarshal( ) was called.

The property can be used in place of the ProDataTable.SetNoSchemaMarshal( ) method. Use this method if you need all DataSet and DataTable parameters in a single proxy method to not pass schema info from .NET to the AppServer.

For example:

```plaintext
RunTimeProperties.NoSchemaMarshal = true;
appObj.getMyDSet(1, out myOutDS);
```

This property applies to the RunTimeProperties class only (it does not apply to Connection objects).

**Note:** This property affects schema marshaling from the .NET Open Client to the AppServer only. To alter marshaling from the AppServer to the client, you can use the -ttmarshal startup parameter, or you can use the temp table properties NO-SCHEMA-MARSHAL and MIN-SCHEMA-MARSHAL. For more information, see OpenEdge Deployment: Startup Command and Parameter Reference and OpenEdge Development: ABL Reference.

**Default:** False
AppServer enhancements

OpenEdge Release 10.2B includes the following enhancements related to AppServers:

- AppServer KeepAlive protocol checks for failed connections
- Changing the AppServer at run time with dynamic properties
- New command-line option for tools: listallprops
- Configuring a default time limit for server applications

AppServer KeepAlive protocol checks for failed connections

When a client machine crashes or when a point in the network route between the server and a client fails, communication stalls and the AppServer may not be aware that a problem exists. Since the AppServer is a shared resource, failures of this type affect all application users. System resources like AppServer agents and memory are tied up and application resources like locked database records are unavailable to other users. It is important to be able to detect these failures and recover resources as quickly as possible.

The AppServer provides the ServerASK messaging protocol to help identify these failures. ASK stands for AppServer KeepAlive and ServerASK denotes that the direction of the keepalive messages is from the server to the clients. (The AppServer does not currently support a reciprocal ClientASK protocol.)

When an AppServer and client are configured to use ServerASK, the AppServer periodically sends a keepalive message to the client and waits for a response. If the client responds within the defined time, then the AppServer knows the connection is valid and resets the count down for the next keepalive message. If the client does not respond in the defined time, then the AppServer presumes there is a failed connection and responds appropriately, usually by disconnecting the socket.

This relationship is defined with three AppServer (ubroker) properties:

- The appServerKeepaliveCapabilities property in the [UBroker.AS] section of the ubroker.properties file enables or disables ServerASK on an AppServer. It is disabled by default.
- The serverASKActivityTimeout property determines the maximum amount of time in seconds from the last activity on a connection that the AppServer waits for a client message before initiating a request.
- The serverASKResponseTimeout property is the maximum amount of time in seconds that the AppServer waits for a response after a request has been issued.

Clients that support ServerASK include:

- ABL client connections
- Java and .NET Open Client connections
- Web Services adapter connections
- SonicMQ Broker Adapter connections
What's New in 10.2B

- SonicESB Invocation Adapter connections
- SonicESB Web Services Adapter connections
- AppServer Internet Adapter connections

Clients of an AppServer with ServerASK enabled participate in the keepalive protocol by default. Each client supports a connection property that allows you to explicitly enable or disable participation in a keepalive relationship with its AppServer.

For complete AppServer administration documentation, see:

| Manual: | OpenEdge Application Server: Administration |

**Changing the AppServer at run time with dynamic properties**

The AppServer and its brokers are presently configured by property settings stored in the ubroker.properties file that ships with the product. Administrators can change these properties before starting the AppServer to customize the server for their environment. Properties are only read at start up and are used to configure the server and its brokers. To change the configuration of the server, an administrator must change the required properties and stop and re-start the AppServer broker.

In Release 10.2B, the AppServer supports run-time changes to some properties. This feature allows users to change some properties at runtime without taking down and re-starting an AppServer broker. The following parts of the AppServer environment now support some dynamic properties:

- NameServer
- Universal Broker
- Appserver agent
- WebSpeed agent
- SonicMQ Broker Adapter

Properties that can be changed at runtime are referred to as dynamic properties. Properties that cannot be changed at runtime are called immutable properties.
New AppServer property

A new property called allowRuntimeUpdates controls whether an AppServer allows runtime updates to properties. It is located in the [UBroker] section of the ubroker.properties file. Table 1–2 describes the valid settings for this property:

Table 1–2: allowRuntimeUpdates property values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Run-time property changes are not allowed for this application server. This is the default value.</td>
</tr>
<tr>
<td>1</td>
<td>Run-time property changes are allowed for this application server. The AppServer listens for changes to the ubroker.properties file and reloads all value and behavioral properties. This is true whether the file is directly edited or changed through the Progress Explorer, OpenEdge Explorer, or any other tool.</td>
</tr>
</tbody>
</table>

Dynamic properties

This section lists properties that are dynamic. The subsections arrange the properties in useful groupings that describe the effect of changing the listed properties at runtime.

The following parts of the AppServer environment support some dynamic properties:

- NameServer
- Universal Broker
- AppServer agent
- WebSpeed agent
- SonicMQ Broker Adapter

General dynamic properties: AppServer and WebSpeed

Any changes to the following properties affect all current and new agents:

- AutoTrimTimeout
- collectStatsData
- flushStatsData
- connectingTimeout
- requestTimeout

General dynamic properties: AppServer, WebSpeed, and DataServer

Any changes to the following properties only affect new agents that are started after these values have been changed because they affect how the agent is started. Existing (running) brokers and agents are not updated. Also note that DataServer agents do not have configuration property files. DataServer agents pick up changed values from new DataServer brokers that do read the properties file.

- srvrExecFile
- srvrMaxPort
What's New in 10.2B

- `srvrMinPort`
- `srvrStartupParam`
- `srvrStartupTimeout`

**General dynamic properties: Appserver agent**

Any changes to the following properties affect all current and new agents:

- `srvrActivateProc`
- `srvrConnectProc`
- `srvrDeactivateProc`
- `srvrDisconnProc`
- `srvrShutdownProc`

Any changes to the following properties only affect new agents that are started after these values have been changed because they affect how the agent is started. Existing (running) agents are not updated:

- `srvrStartupProc`
- `srvrStartupProcParam`

**Logging dynamic properties: Appserver and WebSpeed**

The following properties affect log file information. Any changes will affect all current and new brokers and agents:

- `brkrLoggingLevel`
- `brkrLogEntryTypes`
- `srvrLoggingLevel`
- `srvrLogEntryTypes`

**General dynamic property: AppServer broker and registered NameServer**

Any changes affects all current and new brokers as well as the broker's registered NameServer:

- `priorityWeight`

**General dynamic property: SonicMQ Adapter**

Any changes to the following property affects all current and new broker adapters:

- `srvrLoggingLevel`

**General dynamic properties: NameServer**

Any changes to the following properties affect all current and new NameServers:

- `loggingLevel`
- `logEntryTypes`
- `brokerKeepAliveTimeout`
New command-line option for tools: listallprops

The various command-line tools that display broker information now support a new switch: listallprops. The switch displays all active broker properties. The listed values show the updated values after the update has been made. The following command-line tools support this switch:

• adaptman
• asbman
• wtbsman
• nsman
• oraman
• mssman
• odbman

Configuring a default time limit for server applications

You can specify a default time limit for all remote procedures executed on an AppServer. This property provides a mechanism for configuring an execution time limit for requests submitted by clients that do not support the ability to invoke a timed remote procedure call. This includes the following types of clients:

• Versions of ABL clients from OpenEdge 9 to OpenEdge 10.2A
• OpenClient for Java and OpenClient for .NET
• Web Services Adapter
• Adapter for Sonic ESB

A new property called the srvrExecutionTimeLimit has been added to the [UBroker] section of the ubroker.properties file. This is a non-negative integer property that specifies the maximum time in seconds that a remote procedure may execute on the given AppServer. This timeout value applies to all remote procedures that execute on the AppServer.

The value of zero indicates that no time limit is in effect. The default value of this property is zero, thereby preserving the existing AppServer behavior for applications that do not use this feature.

If the property is set, the information is logged.

Also see the “STOP-AFTER phrase provides timed STOP condition for DO, FOR, and REPEAT blocks” section on page 1–23, and the “CANCEL-REQUESTS-AFTER method provides timed STOP condition for remote procedure calls” section on page 1–24 for additional information.
OpenEdge interceptors for Actional

OpenEdge interceptors are a feature that provides enhanced monitoring capabilities. With Release 10.2B, OpenEdge provides Actional interceptor functionality for most OpenEdge components. The interceptor functionality is built into the OpenEdge components and enabled and configured in OpenEdge with AppServer properties.

Note: Actional Management Server 8.0.3 adds support for using an OpenEdge database as the Actional repository. See the Actional documentation for more information.

For an introduction to Actional, see:

| Manual:                      | OpenEdge Getting Started: Application and Integration Services |

For information about configuring Actional support, see:

| Manual:                      | OpenEdge Application Server: Administration |
OpenEdge Release 10.2B includes the following deployment features:

- Code page conversion of text segments in memory-mapped procedure libraries
- Installing WebClient without Administrator privileges

**Code page conversion of text segments in memory-mapped procedure libraries**

Release 10.2B removes the restriction on memory-mapped procedure libraries that the code page of their text segments must match the AVM’s internal code page (\-cpinternal\). Now, when a text segment in a memory-mapped procedure library has a different code page, the AVM converts the text segment to the AVM’s internal code page in local memory. The AVM loads the converted text segment into the r-code execution buffer. The AVM then swaps the converted text segment to and from the r-code swap file rather than re-reading it from the procedure library.

In previous releases, a code page mismatch of this type generates the following errors at run time:

```
Code page <mmap-codepage> for memory-mapped r-code does not match \-cpinternal\, <\-cpinternal\>. (9073)
*<file> was not found. (293)
```

**Note:** If you specify the Segment Statistics (-yd) startup parameter, the client.mon file now records the swapping of the converted text segments.

Removing this restriction enables you to deploy your application using multiple values for \-cpinternal\ without re-compiling and generating memory-mapped procedure libraries for each code page.

**Installing WebClient without Administrator privileges**

Release 10.2B introduces a new strategy for installing WebClient applications, the “personal installation”, which avoids the need for Administrator-level privileges. In this strategy, you install a separate copy of the WebClient application. The installation uses the HKEY_CURRENT_USER (HKCU) registry hive and your application data directory, such as C:\Documents and Settings\user-name\Application Data.

For more information about installing WebClient without Administrator privileges, see:

**Manual:** OpenEdge Deployment: WebClient Applications
OpenEdge RDBMS

The OpenEdge 10.2B RDMBS includes new and revised features in the following areas:

- **Transparent Data Encryption**
- **Alternate Buffer Pool**

**Transparent Data Encryption**

As part of an overall security strategy, a new security feature, Transparent Data Encryption, is introduced in this release. Transparent Data Encryption provides for data privacy while the data is “at rest” in your OpenEdge database. The Transparent Data Encryption product must be installed in conjunction with an Enterprise Database license. Transparent Data Encryption allows you to specify which data is encrypted and the cipher used to perform the encryption. You also determine how your database is opened, and who has access. Changes to support Transparent Database Encryption include:

- **New required area for your database** — Encryption policies are stored in their own “Encryption Policy Area” that must be added with PROSTRCT before you can enable your database for Transparent Data Encryption.

- **New OpenEdge key store** — A new file, `database-name.ks`, is stored in the same directory as your `database-name.db` file and contains the encryption database master key.

- **New PROUTIL commands**:
  - `ENABLEENCRYPTION` — Enables your database for Transparent Data Encryption.
  - `DISABLEENCRYPTION` — Disables Transparent Data Encryption on your database.
  - `EPOLICY` — Defines, manages, and maintains all aspects of encryption for your database. Use PROUTIL EPOLICY to define and update encryption policies, view status of encryption policies and associated data, encrypt data, perform key store administration, and manage autostart settings.

- **Updates to existing utilities** — Utilities that can manipulate encrypted data, providing a clear-text version of the data, are restricted to authenticated database administrators.

- **Changes to database startup** — To open an encryption-enabled database, you must be authenticated as able to open the database key store that was created when the database was enabled for encryption. Depending on security needs, your database administrator may allow database servers to be autostarted, or may require the `−Passphrase` parameter be added to the server command line to prompt for the key store passphrase. Database utilities and non-networked clients can also add `−Passphrase` to prompt for the key store passphrase. ABL clients that must supply a keystore passphrase, first prompt for it, and then pass it with the `CONNECT` statement by adding `−KeystorePassPhrase`. If you cannot successfully authenticate the key store, the database will not open.

- **Support for Transparent Data Encryption in Auditing** — You can encrypt your audit data, and you can audit changes to your encryption policies and your key store.
• **Support for Transparent Data Encryption in Data Administration** — The Data Administration utility contains a new submenu, Admin ➔ Security ➔ Encryption Policies, which allows you to edit encryption policies, generate encryption keys, and view encryption policy history for objects in Type II storage areas.

• **Support for Transparent Data Encryption in SQL** — The enhanced OpenEdge SQL DDL enables encryption for objects in Type II storage areas. You can define, modify, and rekey encryption specifications for tables, indexes and LOB columns; drop existing encrypted tables, indexes, and LOB columns, and view encryption specifications of tables, indexes, and LOB columns.

• **Support for Transparent Data Encryption in OpenEdge Replication** — You can enable a replicated source database for encryption, and you can replicate an encryption-enabled database.

• **Support for Transparent Data Encryption in OpenEdge Explorer, Progress Explorer, and OpenEdge Management** — You can now do the following:
  – Start a database server for an encryption-enabled database using OpenEdge Explorer, Progress Explorer, or OpenEdge Management if encryption is configured for Autostart, but not if encryption is configured for Manual Start
  – Stop the database server for an encryption-enabled database with either the Autostart configuration or the Manual Start configuration
  – Add an encryption-enabled database that has been started using a script as a scripted database in OpenEdge Explorer or OpenEdge Management

For more information about Transparent Data Encryption, see:

| Manuals: | OpenEdge Getting Started: Core Business Services  
|          | OpenEdge Data Management: Database Administration  
|          | OpenEdge Development: Basic Database Tools |
| Online help: | Database Admin Help |
Alternate Buffer Pool

The Alternate Buffer Pool is a collection of buffers in shared memory that are logically separate from the primary buffer pool. Management of the buffers in the Alternate Buffer Pool is independent of the primary buffer pool. Assigning specific database areas or objects to occupy buffers in the Alternate Buffer Pool may improve your buffer hit rate, thereby reducing the need to read and write buffers to and from disk, possibly improving performance. Changes to support the Alternate Buffer Pool include:

- New PROUTIL commands that you can use to assign areas to the Alternate Buffer Pool, remove areas, and view object and area assignment
  - ENABLEB2
  - DISABLEB2
  - VIEWB2
- Support for managing objects in Type II areas in the Alternate Buffer Pool through Data Administration
- SQL DDL extensions for managing objects in Type II areas in the Alternate Buffer Pool
- Updates to Virtual System Tables and PROMON screens to monitor Alternate Buffer Pool status
- New Blocks in Alternate Buffer Pool (\(-B2\)) startup parameter to specify the size of the Alternate Buffer Pool
- Support for increasing the size of the Alternate Buffer Pool with PROUTIL INCREASETO

You can configure objects and areas to use the Alternate Buffer Pool with any database license; however, allocating the Alternate Buffer Pool with the \(-B2\) startup parameter requires an Enterprise Database license.

For more information about the Alternate Buffer Pool, see:

<table>
<thead>
<tr>
<th>Manuals</th>
<th>OpenEdge Data Management: Database Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: Basic Database Tools</td>
</tr>
<tr>
<td>Online help</td>
<td>Database Administration Utilities Help</td>
</tr>
</tbody>
</table>
OpenEdge Replication

OpenEdge Replication in Release 10.2B provides support for the following tasks related to database encryption:

- Enabling Transparent Data Encryption for an OpenEdge Replication-enabled source database
- Enabling OpenEdge Replication for a database that is currently enabled for Transparent Data Encryption

For more information about OpenEdge Replication, see:

| Manual: | OpenEdge Replication: User Guide |
OpenEdge Management and OpenEdge Explorer

OpenEdge Release 10.2B includes OpenEdge Management and OpenEdge Explorer enhancements in the following areas:

- Installation and packaging
- Support for alternate buffer pools
- Support for dynamic property updates
- Support for Actional monitoring of OpenEdge resources
- Support for additional database properties
- Support for automatic raising of STOP on time-out

Installation and packaging

OpenEdge Explorer is now packaged with OpenEdge. OpenEdge Explorer is installed with the same products that include a Progress Explorer installation.

OpenEdge Explorer and OpenEdge Management are packaged with OpenEdge and no longer require a separate installation; however, OpenEdge Management still requires a separate license. When OpenEdge Management is licensed, the OpenEdge Explorer functionality is included.

Support for alternate buffer pools

Support for alternate buffer pools is available in OpenEdge Management and OpenEdge Explorer. Property configuration—through the **Alternate buffer pool** property—is supported for managed databases in both OpenEdge Management and OpenEdge Explorer. OpenEdge Management also provides support for polling, trending and graphing.

An Enterprise database license is required, and you cannot set alternate buffer pools on secondary brokers.

For more information about support for alternate buffer pools, see:

| Manual: | OpenEdge Management and OpenEdge Explorer: Configuration |

Support for dynamic property updates

OpenEdge Management and OpenEdge Explorer now include a new **Enable dynamic property updates** configuration property (named **allowRuntimeUpdates** in ubroker.properties) for AppServer, WebSpeed, NameServer, and SonicMQ Adapter instances. You can enable this property to allow dynamic changes to several of the specific instance’s properties, without having to shut down the instance.
For more information about support for dynamic property updates, see:

Manual:  *OpenEdge Management and OpenEdge Explorer: Configuration*

### Support for Actional monitoring of OpenEdge resources

OpenEdge Release 10.2B includes support for Actional monitoring of certain OpenEdge resources. As part of this support, OpenEdge Management and OpenEdge Explorer now include the following two new configuration properties for the AppServer Internet Adapter (AIA), the AppServer, WebSpeed, and the SonicMQ Adapter:

- **Enable Actional Monitoring** — Enable this option to configure Actional instrumentation
- **Actional Group Name** — Optionally sets the Group value that Actional uses in creating flow maps

For more information about OpenEdge Management and OpenEdge Explorer support for Actional monitoring, see:

Manual:  *OpenEdge Management and OpenEdge Explorer: Configuration*
Support for additional database properties

Support for the database properties in the following table has been added to the management console for OpenEdge Management and OpenEdge Explorer in 10.2B:

<table>
<thead>
<tr>
<th>After-image archival directory list</th>
<th>Create after-image archival directories</th>
<th>After-image archival interval</th>
<th>Base index number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base table number</td>
<td>Database service communication area size</td>
<td>Windows Event Level</td>
<td>Group delay</td>
</tr>
<tr>
<td>Minimum heap size</td>
<td>Index range size</td>
<td>Lock table hash table size</td>
<td>Maximum number of JTA transactions</td>
</tr>
<tr>
<td>Maximum private database buffers per user</td>
<td>Maximum number of database areas</td>
<td>Nap time increment</td>
<td>Nap time steps between nap time</td>
</tr>
<tr>
<td>Pending connect time-out</td>
<td>Alternate buffer pool</td>
<td>Number of open SQL cursors</td>
<td>SQL stack size</td>
</tr>
<tr>
<td>SQL statement cache size</td>
<td>SQL sorting memory</td>
<td>SQL sorting on disk</td>
<td>SQL temporary data page size</td>
</tr>
<tr>
<td>Disable SSL session cache</td>
<td>Enable SSL for remote connections</td>
<td>Key alias name</td>
<td>Key alias password</td>
</tr>
<tr>
<td>Pin shared memory segments</td>
<td>Maximum shared-memory segment size</td>
<td>Storage object cache size</td>
<td>TXE commit lock skip limit</td>
</tr>
<tr>
<td>Table range size</td>
<td>User MUX latches</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

For more information about configuring these database properties, see:

Manual: [OpenEdge Management and OpenEdge Explorer: Configuration](#)

Support for automatic raising of STOP on time-out

Support for automatic raising of STOP on time-out has been added to OpenEdge Management and OpenEdge Explorer. The AppServer Agent configuration tab now includes a new property named **Execution Time Limit**. The value you type for this property sets the maximum time in seconds that a remote procedure can execute on a given AppServer and applies to all remote procedures that execute on the AppServer.

For more information about support for this feature, see:

Manual: [OpenEdge Management and OpenEdge Explorer: Configuration](#)
DataServers

OpenEdge Release 10.2B includes these DataServers features:

- Enhanced logging support for DataServer for MS SQL Server
- DataServer support for datetime data types
- Support for BLOB data type in DataServer for MS SQL Server
- Support for RECID in DataServer for MS SQL Server
- Support for RECID in DataServer for ODBC
- Support for default values in DataServer for ODBC

Enhanced logging support for DataServer for MS SQL Server

In OpenEdge Release 10.2B, logging support for the DataServer for MS SQL Server includes the enhanced logger infrastructure previously available only to OpenEdge clients and AppServer agents. Enhanced Logging for DataServers provides capabilities for reporting and diagnostics from both client log parameters and server-side -Dsrv options. Enhancements include new log entry types, the ability to specify logging levels for each entry type, and other miscellaneous log file options.

For more information about DataServer for MS SQL Server, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help Data Dictionary online help</td>
</tr>
</tbody>
</table>

DataServer support for datetime data types

OpenEdge DataServer for MS SQL Server is enhanced to support DATETIME-TZ data types and other datetime data types supported in MS SQL Server 2008, some which have higher precision. Datetime data types enhance the DataServer’s ability to store and manipulate date and time data and to accurately manage data that spans time zones.

For more information about DataServer for MS SQL Server, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help Data Dictionary online help</td>
</tr>
</tbody>
</table>
Support for BLOB data type in DataServer for MS SQL Server

DataServer for MS SQL Server is enhanced to use the OpenEdge BLOB data type, enabling you to handle data records of up to 1 gigabyte in size. Release 10.2B provides the following enhancements:

- The ability to migrate an OpenEdge database with BLOB data type to a MS SQL Server database as VARBINARY (MAX) and pull back as BLOB in the schema holder
- The ability to pull MS SQL Server data types VARBINARY (MAX), IMAGE, or VARBINARY (MAX) FILESTREAM as an OpenEdge BLOB data type into the schema holder
- The ability to read or write data to or from the MS SQL Server database through the ABL BLOB data type by using the ABL COPY-LOB operation

You are only able to perform an entire BLOB read and write operation through the COPY-LOB statement. COPY-LOB capabilities for the MS SQL Server DataServer exclude the use of offset operations via the STARTING AT, OVERLAY, and FOR copy phrases.

For more information about DataServer for MS SQL Server, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary online help</td>
</tr>
</tbody>
</table>

Support for RECID in DataServer for MS SQL Server

In OpenEdge Release 10.2B, OpenEdge DataServer for MS SQL Server enhances RECID support for tables migrated to an MS SQL Server data source. As in past releases, you can use the Data Dictionary to enable RECID creation by means of database triggers. Beginning in Release 10.2B, you can also enable RECID creation using MS SQL Server computed column technology. These options can be selected using the Data Dictionary DataServer migration tool. Enabling RECID creation through database triggers remains the default behavior.

While both options are offered to support backward compatibility with previous releases, the newer computed column option is recommended as it offers more robust performance for record creation over the trigger-based solution. Please note that the computed column option is available only when using MS SQL Server 2005 and later releases.

For more information about DataServer for MS SQL Server, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary online help</td>
</tr>
</tbody>
</table>
Support for RECID in DataServer for ODBC

In OpenEdge Release 10.2B, OpenEdge DataServer for ODBC enhances RECID support for tables migrated to a DB2/400 foreign data source. Previously, this support was provided only to Sybase databases.

For more information about DataServer for ODBC, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for ODBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary online help</td>
</tr>
</tbody>
</table>

Support for default values in DataServer for ODBC

In OpenEdge Release 10.2B, DataServer for ODBC is enhanced to allow default values defined for fields in an OpenEdge database to be pushed to a foreign database during migration. This feature is available only when migrating to DB2/400 databases.

For more information about DataServer for ODBC, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: DataServer for ODBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Data Administration online help</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary online help</td>
</tr>
</tbody>
</table>
OpenEdge SQL

In Release 10.2B, OpenEdge SQL Data Definition Language (DDL) is enhanced to support both alternate database buffer pools and Transparent Data Encryption.

Alternate buffer pools are managed for two purposes:

- To improve database performance
- To enable database administrators to enhance encryption performance

The enhanced OE SQL DDL enables encryption. Specifically, DBAs can:

- Define encryption specifications for new or existing tables, indexes, and LOB columns
- Modify encryption/decryption specifications for existing tables, indexes, and LOB columns
- Drop existing encrypted tables, indexes, and LOB columns
- View encryption specifications of tables, indexes, and LOB columns
- Rekey existing encrypted tables, indexes and LOB columns

For more information about OpenEdge SQL, see:

| Manuals: | OpenEdge Data Management: SQL Development  
|          | OpenEdge Data Management: SQL Reference |
10.2B Feature Comparisons

This chapter provides comparisons of how features in Release 10.2B differ from previous releases in the following product areas:

- XREF-XML option output from the ABL compiler
- Compiler errors on string attributes
- ORIGIN-HANDLE copied during temp-table and ProDataSet copy operations
- Generalized serialization behavior affects XML serialization
- Changes to ProxyGen .NET tab
- Changing AppServer properties at runtime
- OpenEdge Management and OpenEdge Explorer
XREF-XML option output from the ABL compiler

The XREF option output on the COMPILE statement includes new and updated reference types. For more information, see the “ABL Compiler XREF output” section on page 1–18.

If you use the XREF-XML option, especially to generate output that you parse with computer code, note the following changes:

- A new <Is-abstract> boolean</Is-abstract> element has been added to each reference type node in the XML output, where boolean is true if the reference type identifies an abstract class, property, method, or event reference, and is false otherwise. The sample XREF-XML output shown in the COMPILE statement reference entry includes this new element.

- The properties\schemas folder under your OpenEdge installation directory contains a new version of the XML Schema file (xrefd0003.xsd), which describes the complete schema for the XREF-XML option output for Release 10.2B.
Compiler errors on string attributes

With Release 10.2B, ABL support for .NET generic types also supports static type-name references using generic type names. For information on ABL support for .NET generic types, see the “.NET generic types” section on page 1–17. Because generic type names must be enclosed in quotes, static generic type-name references have the same syntax as references to string attributes. For example, where `ReferenceEquals()` is a static method on the "System.Collections.Generic.List<INTEGER>" class, the following DISPLAY statements have a similar syntax for displaying the result of calling the static method and displaying a character string with a string attribute:

```
DISPLAY "Customer":L50.
```

In Release 10.2A, because a quoted string followed by a colon (:) is always a reference to a string attribute, if such an attribute is invalid, the ABL compiler returns the following error message:

```
Invalid string attribute (2938)
```

However, in Release 10.2B, because static generic type-name references appear much like string attribute references, if you do code an invalid string attribute, the ABL compiler attempts to determine if it is really a reference to a static member of a generic class. If it is not such a reference, the ABL compiler returns a less informative compiler error message like this:

```
Unknown Field or Variable name - Customer (201)
```
ORIGIN-HANDLE copied during temp-table and ProDataSet copy operations

The ORIGIN-HANDLE attribute returns the handle of the temp-table in the original source ProDataSet object that corresponds to the temp-table currently associated with the current temp-table handle. The AVM uses this value to match up temp-tables in a MERGE-CHANGES operation.

For Release 10.2B the ORIGIN-HANDLE value is copied from the source temp-table to the target temp-table in all cases for the following methods:

- COPY-DATASET( ) method (ProDataSet object handle)
- COPY-TEMP-TABLE( ) method (Temp-table object handle)

Prior to 10.2B, ORIGIN-HANDLE was only copied from the source to the target if the source had before-table records. This behavior was not documented.

You do not have to manually copy ORIGIN-HANDLE from the source temp-table to the target temp-table, since the AVM automatically does this. The following code fragment (in bold) is no longer necessary:

```plaintext
hDSChanges: COPY-DATASET(hDSparent).
DO iBuffer = 1 TO hDSparent: NUM-BUFFERS:
   hDSChanges: GET-BUFFER-HANDLE(iBuffer): TABLE-HANDLE: ORIGIN-HANDLE =
   hDSparent: GET-BUFFER-HANDLE(iBuffer): TABLE-HANDLE: ORIGIN-HANDLE.
END.
```
Generalized serialization behavior affects XML serialization

OpenEdge Release 10.2B includes some generalized serialization behavior that affects the existing functionality for serializing data into XML.

ABL now has pairs of options/attributes with overlapping purposes for how the AVM outputs XML data:

- SERIALIZE-HIDDEN and XML-NODE-TYPE
- SERIALIZE-NAME and XML-NODE-NAME

**Note:** The following descriptions apply equally to the options as well as to the attributes.

The XML-NODE-TYPE attribute’s purpose overlaps with the SERIALIZE-HIDDEN attribute. Because of this overlap, the attributes interact as follows:

- The WRITE-XML( ) method always uses the XML-NODE-TYPE attribute value. If you set a value for the XML-NODE-TYPE attribute, it keeps that value regardless of how you set the SERIALIZE-HIDDEN attribute.
- If you do not set the XML-NODE-TYPE attribute and set the SERIALIZE-HIDDEN attribute to TRUE, the AVM sets XML-NODE-TYPE to "HIDDEN".
- If you do not set the XML-NODE-TYPE attribute and set the SERIALIZE-HIDDEN attribute to FALSE, the AVM sets XML-NODE-TYPE to "ELEMENT", as was the default behavior in previous releases.
- If you do not set either attribute, the AVM sets XML-NODE-TYPE to "ELEMENT" and sets SERIALIZE-HIDDEN to FALSE.

The XML-NODE-NAME attribute’s purpose overlaps with the SERIALIZE-NAME attribute. Because of this overlap, the attributes interact as follows:

- The READ-XML( ) and WRITE-XML( ) methods always use the XML-NODE-NAME attribute value. If you set a value for the XML-NODE-NAME attribute, it keeps that value regardless of how you set the SERIALIZE-NAME attribute.
- If you do not set the XML-NODE-NAME attribute and set the SERIALIZE-NAME attribute, the AVM sets XML-NODE-NAME equal to SERIALIZE-NAME.
- If you do not set either attribute, the AVM sets both to the ABL object name.
Changes to ProxyGen .NET tab

ProxyGen now allows another choice for .NET assemblies (DLLs). To accommodate this change, the UI on the ProxyGen .NET tab has changed as follows:

- The **Use Strongnamed Runtime** checkbox has been replaced with a new **Runtime** combobox. The **Digitally Signed** selection is equivalent to the unchecked state of the previous checkbox. The **Strongnamed Signed** selection is equivalent to the old checked state. The **Strongnamed** selection is new for this release.

- The **Delay Sign** checkbox now generates an error message if it is selected when the **Runtime** combobox is set to **Digitally Signed**.

See the “.NET Open Client supports strongly-named, unsigned assemblies” section on page 1–26 for a description of the new assembly option.
DataServer for MS SQL Server support for datetime data types

The DataServer for MS SQL Server is enhanced in Release 10.2B to support DATETIME-TZ data types and other datetime data types supported by MS SQL Server 2008.

DataServer for MS SQL Server introduced support for ABL DATETIME data type in OpenEdge Releases 10.1C and 10.2A. However, support for the ABL DATETIME-TZ data type was not implemented since MS SQL Server releases, at that time, did not support datetime data types with a time zone component.

MS SQL Server 2008 introduces several date and time data types, including a datetime data type with a time zone component. OpenEdge 10.2B is enhanced to support these new data types.

DataServer schema holder version compatibility

Occasionally information is added to the schema holder in certain OpenEdge DataServer versions for new or changed features that create version incompatibilities across schema holders. If you create a schema holder using a version of the OpenEdge dictionary that includes a feature without backward compatibility and then try to use that new feature with an older version of the product, the DataServer client may receive an incompatibility message using the newer schema. Generally speaking, if the new feature is not used in the schema holder built with the newer version of the OpenEdge dictionary, then the schema holder should typically continue to have backward compatibility.

The schema holder changes in the specified releases may cause schema holders with those features to be incompatible with older versions of the DataServer. The following compatibility issues currently exist:

- ODBC schema holders in Releases 9.1e03, 10.0b04 and 10.1A - replace "##" in extent column naming convention with "__" for DB2
- ODBC and MSS schema holders in Release 10.1B - add 64-bit RECID's
- MSS schema holders in Release 10.2B - add RECID's based on computed columns
Changing AppServer properties at runtime

Several existing ubroker properties can now be changed at runtime. For complete information on this new ability, see the “Changing the AppServer at run time with dynamic properties” section on page 1–30.
OpenEdge Management and OpenEdge Explorer

The following changes exist in OpenEdge Management and/or OpenEdge Explorer Release 10.2B:

- Starting OpenEdge Management or OpenEdge Explorer from the OpenEdge program folder
- Using either the unglue or reglue script
- Using Proenv in place of Set OpenEdge Management Environment

Starting OpenEdge Management or OpenEdge Explorer from the OpenEdge program folder

OpenEdge Management (if licensed) and OpenEdge Explorer are now installed with OpenEdge. To start OpenEdge Management/OpenEdge Explorer from the Windows Desktop on your local host, select Start → Programs → OpenEdge → Management Console. If you have only OpenEdge Explorer installed, select Start → Programs → OpenEdge → OpenEdge Explorer.

Using either the unglue or reglue script

The unglue and reglue scripts used in previous releases of OpenEdge Management have been modified. On UNIX platforms, the unglue script previously allowed you to unglue OpenEdge Management from OpenEdge before uninstalling it, and the reglue script allowed you to reglue OpenEdge Management to an installation of OpenEdge.

You now run the unglue script to unglue OpenEdge Management or OpenEdge Explorer from OpenEdge before you can enable a machine as a remote container. When you run the unglue script on the intended remote container, you are disabling (but not uninstalling) OpenEdge Management or OpenEdge Explorer in the OpenEdge product.

Should you later decide to discontinue using a machine as a remote container, you can run reglue, which re-enables OpenEdge Management or OpenEdge Explorer in the OpenEdge product.

For more information about using unglue and reglue, see:

Manual: OpenEdge Management and OpenEdge Explorer: Getting Started
Using Proenv in place of Set OpenEdge Management Environment

You can execute both OpenEdge Management and OpenEdge command-line utilities by using the Proenv utility instead of the Set OpenEdge Management environment window. To access Proenv, choose Start → Programs → OpenEdge → Proenv.

For more information about using the Proenv utility with OpenEdge Management or OpenEdge Explorer commands, see:

| Manual: | OpenEdge Management and OpenEdge Explorer: Getting Started |
This chapter describes the product documentation provided for OpenEdge Release 10.2B. It includes the following sections:

- Changes to the documentation set
- Release 10.2B documentation set
- Accessing OpenEdge Documentation
Changes to the documentation set

The OpenEdge Release 10.2B documentation set includes the following changes:

- **OpenEdge Management and OpenEdge Explorer: Installation** is no longer available, since the OpenEdge Management and OpenEdge Explorer installs are now part of the core OpenEdge installation.

- The OpenEdge Management and OpenEdge Explorer documentation PDFs are no longer included with the product. Documentation is available:
  - In PDF format in the doc_samples folder on the product DVD provided with OpenEdge.
  - In PDF and HTML format from the following location:
    - From the Help → Documentation link in the management console.
    - From the context-sensitive Help icon (the question mark) in the detail control bar of the management console.

- **OpenEdge Development: ABL Handbook** was removed from the OpenEdge documentation set. The ABL Handbook in its current form is available on the Progress Communities web site: [http://communities.progress.com/pcom/docs/DOC-16074](http://communities.progress.com/pcom/docs/DOC-16074).

- Added a new manual, OpenEdge Getting Started: ABL Essentials. The content was extracted from OpenEdge Development: ABL Handbook and migrated to this manual.

- Added a new manual, OpenEdge Getting Started: GUI for .NET Primer.

- Added a new manual, OpenEdge Development: Working with JSON.

- Added a new web paper, Actional Interceptors for OpenEdge.

- Renamed OpenEdge Getting Started: Object-oriented Programming to OpenEdge Development: Object-oriented Programming.

- Renamed Dynamic Invoke web paper to Dynamic Call Object.

- Added a series of video presentations on a number of OpenEdge Architect Getting Started topics, Visual Designer topics, and GUI for .NET topics. These videos are accessible from the Architect online help, from the Architect Welcome page, and from the Progress Communities web site: [http://communities.progress.com/pcom/docs/DOC-101140](http://communities.progress.com/pcom/docs/DOC-101140).
## Release 10.2B documentation set

Table 3–1 lists the 10.2B documentation available for each technology area. You can access the entire OpenEdge 10.2B product documentation set from the following Web site: http://communities.progress.com/pcom/docs/DOC-16074.

### Table 3–1: 10.2B documentation map arranged by technology

<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
</table>
| Application Development with ABL | OpenEdge Getting Started: ABL Essentials  
OpenEdge Development: ABL Reference  
OpenEdge Development: Debugging and Troubleshooting  
OpenEdge Development: Error Handling  
OpenEdge Development: Object-oriented Programming  
OpenEdge Development: ProDataSets  
OpenEdge Development: Programming Interfaces  
Dynamic Call Object web paper |
| ABL Application Development with the OpenEdge GUI for .NET | OpenEdge Getting Started: GUI for .NET Primer  
OpenEdge Development: GUI for .NET Mapping Reference  
OpenEdge Development: GUI for .NET Programming |
| Error handling | OpenEdge Development: Error Handling |
| Auditing | OpenEdge Getting Started: Core Business Services  
OpenEdge Data Management: Database Administration  
OpenEdge Development: Programming Interfaces |
| Business Intelligence and Reporting | OpenEdge Reporting: Query/Results Administration and Development  
OpenEdge Reporting: Query/Results for UNIX  
OpenEdge Reporting: Query/Results for Windows  
OpenEdge Reporting: Deploying Crystal Reports  
OpenEdge Reporting: Report Builder Deployment |
| Data Management | OpenEdge Getting Started: Database Essentials  
OpenEdge Data Management: Database Administration  
OpenEdge Data Management: DataServer for Microsoft SQL Server  
OpenEdge Data Management: DataServer for ODBC  
OpenEdge Data Management: DataServer for Oracle  
OpenEdge Data Management: SQL Development |
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment</td>
<td><em>OpenEdge Deployment: Managing ABL Applications</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Deployment: Startup Command and Parameter Reference</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Deployment: WebClient Applications</em></td>
</tr>
<tr>
<td>Installation and configuration</td>
<td><em>OpenEdge Getting Started: Installation and Configuration</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Deployment: Startup Command and Parameter Reference</em></td>
</tr>
<tr>
<td>Internationalization and localization</td>
<td><em>OpenEdge Development: Internationalizing Applications</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Translation Manager</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Visual Translator</em></td>
</tr>
<tr>
<td>OpenEdge Architect</td>
<td><em>OpenEdge Getting Started: Introducing the OpenEdge Architect Visual Designer</em></td>
</tr>
<tr>
<td></td>
<td>OpenEdge Architect online help</td>
</tr>
<tr>
<td>OpenEdge Management and OpenEdge Explorer</td>
<td><em>OpenEdge Replication: User Guide</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management and OpenEdge Explorer: Getting Started</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management and OpenEdge Explorer: Configuration</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Database Management</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Resource Monitoring</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Servers, DataServers, Messengers, and Adapters</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Alerts Guide and Reference</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Trend Database Guide and Reference</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Management: Reporting</em></td>
</tr>
<tr>
<td>OpenEdge Replication</td>
<td><em>OpenEdge Replication: User Guide</em></td>
</tr>
<tr>
<td>OpenEdge Studio</td>
<td><em>OpenEdge Getting Started: Progress OpenEdge Studio</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: AppBuilder</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Basic Database Tools</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Basic Development Tools</em></td>
</tr>
</tbody>
</table>
### Table 3–1: 10.2B documentation map arranged by technology

<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progress Dynamics</td>
<td><em>OpenEdge Development: Progress Dynamics Getting Started</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Basic Development</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Administration</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Advanced Development</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Managers API Reference</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Repository Reference</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Progress Dynamics Web Development Guide</em></td>
</tr>
<tr>
<td>Open Client interfaces to a Java or .NET application or a Web service definition</td>
<td><em>OpenEdge Development: Open Client Introduction and Programming</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Java Open Clients</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: .NET Open Clients</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Web Services</em></td>
</tr>
<tr>
<td>Security</td>
<td><em>OpenEdge Getting Started: Core Business Services</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Programming Interfaces</em></td>
</tr>
<tr>
<td>Service Oriented Architectures (SOA)</td>
<td><em>OpenEdge Getting Started: Application and Integration Services</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Application Server: Administration</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Application Server: Developing AppServer Applications</em></td>
</tr>
<tr>
<td>SmartObjects and ADM</td>
<td><em>OpenEdge Development: ADM and SmartObjects</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: ADM Reference</em></td>
</tr>
<tr>
<td>Sonic messaging and application integration environment</td>
<td><em>OpenEdge Development: Messaging and ESB</em></td>
</tr>
<tr>
<td>WebSpeed</td>
<td><em>OpenEdge Getting Started: WebSpeed Essentials</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Application Server: Developing WebSpeed Applications</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Working with JSON</em></td>
</tr>
<tr>
<td>Working with JSON and XML</td>
<td><em>OpenEdge Development: Working with JSON</em></td>
</tr>
<tr>
<td></td>
<td><em>OpenEdge Development: Working with XML</em></td>
</tr>
</tbody>
</table>
Accessing OpenEdge Documentation

You can install the OpenEdge Release 10.2B documentation from the Documentation and Samples directory (doc_samples) on the OpenEdge product DVD (or from the Product Download Center).

You can also access the most recent version of the OpenEdge Release 10.2B documentation from the Progress Communities web site: