OpenEdge Development:
Basic Development Tools
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>1–1</td>
</tr>
<tr>
<td><strong>1. Application Development Environment</strong></td>
<td>1–1</td>
</tr>
<tr>
<td>Starting the ADE tools</td>
<td>1–2</td>
</tr>
<tr>
<td>Procedure Editor</td>
<td>1–2</td>
</tr>
<tr>
<td>Application Compiler</td>
<td>1–3</td>
</tr>
<tr>
<td>Accessing menus and menu options</td>
<td>1–4</td>
</tr>
<tr>
<td>Navigating in dialog boxes and windows</td>
<td>1–5</td>
</tr>
<tr>
<td>Using the Tools menu</td>
<td>1–6</td>
</tr>
<tr>
<td>Tools&gt;Procedure Editor</td>
<td>1–6</td>
</tr>
<tr>
<td>Tools&gt;Data Dictionary</td>
<td>1–6</td>
</tr>
<tr>
<td>Tools&gt;OS Shell</td>
<td>1–6</td>
</tr>
<tr>
<td>Tools&gt;Application Compiler</td>
<td>1–6</td>
</tr>
<tr>
<td>Using the Help menu</td>
<td>1–7</td>
</tr>
<tr>
<td>Help&gt;Messages</td>
<td>1–7</td>
</tr>
<tr>
<td>Help&gt;Recent Messages</td>
<td>1–7</td>
</tr>
<tr>
<td>Help&gt;Keyboard</td>
<td>1–7</td>
</tr>
<tr>
<td>Help&gt;About Tool</td>
<td>1–7</td>
</tr>
<tr>
<td><strong>2. Procedure Editor Tasks</strong></td>
<td>2–1</td>
</tr>
<tr>
<td>Starting the Procedure Editor</td>
<td>2–2</td>
</tr>
<tr>
<td>Using edit buffers</td>
<td>2–4</td>
</tr>
<tr>
<td>Listing open buffers</td>
<td>2–5</td>
</tr>
<tr>
<td>Switching buffers</td>
<td>2–5</td>
</tr>
<tr>
<td>Displaying buffer information</td>
<td>2–5</td>
</tr>
<tr>
<td>Working with procedures</td>
<td>2–6</td>
</tr>
<tr>
<td>Creating a new procedure</td>
<td>2–6</td>
</tr>
<tr>
<td>Opening a procedure</td>
<td>2–6</td>
</tr>
<tr>
<td>Saving a procedure</td>
<td>2–7</td>
</tr>
<tr>
<td>Printing a procedure</td>
<td>2–8</td>
</tr>
<tr>
<td>Closing a procedure</td>
<td>2–8</td>
</tr>
</tbody>
</table>
Figures

Figure 2–1: Procedure Editor window .................................................. 2–2
Figure 2–2: Buffer List dialog box ....................................................... 2–5
Figure 4–1: Procedure Editor menu bar ................................................. 4–2
Figure 4–2: Open dialog box ................................................................. 4–3
Figure 4–3: Files dialog box ................................................................. 4–4
Figure 4–4: Close alert dialog box ....................................................... 4–4
Figure 4–5: Save As dialog box ............................................................. 4–5
Figure 4–6: Save Buffers with Changes dialog box ............................... 4–6
Figure 4–7: Insert File dialog box ......................................................... 4–8
Figure 4–8: Field Selector dialog box .................................................. 4–8
Figure 4–9: Find dialog box ................................................................. 4–9
Figure 4–10: Replace dialog box .......................................................... 4–10
Figure 4–11: Goto Line dialog box ....................................................... 4–11
Figure 4–12: Buffer List dialog box ..................................................... 4–12
Figure 4–13: Buffer Information dialog box ........................................ 4–13
Figure 4–14: Compiled procedure example output to screen ................ 4–15
Figure 4–15: Compiler Messages dialog box ...................................... 4–16
Figure 5–1: Application Compiler window .......................................... 5–3
Figure 5–2: File Specification dialog box ............................................. 5–6
Figure 5–3: Compiler Results dialog box ............................................. 5–7
Figure 5–4: Compiler Options dialog box ........................................... 5–8
Tables

Table 1–1: Tools menu .................................................. 1–6
Table 1–2: Help menu ...................................................... 1–7
Table 2–1: Procedure editor features .................................. 2–3
Table 3–1: File-specific events ............................................ 3–4
Table 3–2: Usage scenario for the Procedure Editor .................. 3–6
Table 4–1: Procedure Editor menus ....................................... 4–2
Table 4–2: File menu ......................................................... 4–3
Table 4–3: Files dialog box elements ...................................... 4–4
Table 4–4: Save As dialog box fields ...................................... 4–5
Table 4–5: Edit menu ......................................................... 4–7
Table 4–6: Field Selector dialog box elements ......................... 4–8
Table 4–7: Search menu ...................................................... 4–9
Table 4–8: Find dialog box fields .......................................... 4–9
Table 4–9: Replace dialog box elements ................................... 4–10
Table 4–10: Buffer menu .................................................... 4–12
Table 4–11: Buffer Information dialog box elements .................. 4–13
Table 4–12: Compile menu .................................................. 4–14
Table 5–1: Save New .r toggle box and the compilation process ....... 5–4
Table 5–2: Compiler action buttons ....................................... 5–5
Table 5–3: File Specification dialog box elements ..................... 5–6
Table 5–4: Application Compiler menu ................................... 5–6
Table 5–5: File menu ......................................................... 5–7
Table 5–6: Compile menu ..................................................... 5–7
Table 5–7: Options menu .................................................... 5–8
Table 5–8: Compiler Options dialog box fields ......................... 5–8
This Preface contains the following sections:

- Purpose
- Audience
- Organization
- Using this manual
- Typographical conventions
- OpenEdge messages
- Third party acknowledgements
Purpose

This book is a user guide for OpenEdge® Release 10 basic development toolset for character mode. These tools include the OpenEdge Procedure Editor, the Data Dictionary, and the OpenEdge Application Compiler.

Audience

This book is intended for developers who want to use OpenEdge basic development tools to develop their applications.

Organization

Chapter 1, “Application Development Environment”

Describes how to access the Application Development Environment (ADE), access each OpenEdge tool, and use on-line help for error messages.

Chapter 2, “Procedure Editor Tasks”

Describes how to access the OpenEdge Procedure Editor and use it to perform tasks. The OpenEdge Procedure Editor allows you to create, write, compile, and run ABL procedures.

Chapter 3, “Procedure Editor Integration Hooks”

Describes the integration hooks you can add to the OpenEdge Procedure Editor.

Chapter 4, “Procedure Editor Reference”

Describes the OpenEdge Procedure Editor menu options and dialog boxes.

Chapter 5, “Application Compiler”

Describes how to access the Application Compiler and use its menu options. The Application Compiler allows you to compile a set of source procedures and create ABL r-code.
Using this manual

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


References to ABL compiler and run-time features

ABL is both a compiled and 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.”

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><em>SMALL, BOLD CAPITAL LETTERS</em></td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
<tr>
<td><strong>KEY1+KEY2</strong></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><strong>KEY1 KEY2</strong></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>
</tbody>
</table>
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.

---

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>Fixed-width italics indicate variables in syntax statements.</td>
</tr>
<tr>
<td>Fixed-width italics</td>
<td>Fixed-width bold indicates variables with special emphasis.</td>
</tr>
<tr>
<td>Fixed-width bold</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>UPPERCASE fixed width</td>
<td></td>
</tr>
<tr>
<td>![multi-step procedure icon]</td>
<td>This icon (three arrows) introduces a multi-step procedure.</td>
</tr>
<tr>
<td>![single-step procedure icon]</td>
<td>This icon (one arrow) introduces a single-step procedure.</td>
</tr>
</tbody>
</table>
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**

On 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 software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xerces C++ Parser (XML) and Xerces2 Java Parser (XML)); Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces Parser (XML)); and Copyright © 2000-2003 The Apache Software Foundation. All rights reserved (Ant). The names “Apache,” “Xerces,” “ANT,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact apache@apache.org. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.


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

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

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

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

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

OpenEdge includes 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 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 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 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 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 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 Sonic software which includes a version of the Saxon XSLT and XQuery Processor from Saxonica Limited that has been modified by Progress Software Corporation. The contents of the Saxon source code and the modified source code file (Configuration.java) 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 (MPL-1.0.html) can also be found in the installation directory, in the Docs7.5/third_party_licenses folder, along with a copy of the modified code (Configuration.java); and a description of the modifications can be found in the Progress SonicMQ and Progress Sonic ESB v7.5 README files. 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 The SAXON XSLT and XQuery Processor from Saxonica Limited. The Initial Developer of the Original Code is Michael Kay (http://www.saxonica.com/products.html). Portions created by Michael Kay are Copyright © 2001-2005. All rights reserved. Portions created by Progress Software Corporation are Copyright © 2007. All rights reserved.

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

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 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 (decrsttod.c; strtodg.c); Copyright © 1998, 2000 by Lucent Technologies All Rights Reserved (decrstrof.c; strtord.c); Copyright © 1998 by Lucent Technologies All Rights Reserved (dmisc.c; gdtola.h; gethex.c; gmisc.c; sum.c); Copyright © 1998, 1999 by Lucent Technologies All Rights Reserved (gdtota.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). 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 OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

OpenEdge includes Infragistics NetAdvantage for .NET v2008 Vol 1 CLR 2.0. Copyright © 1996-2008 Infragistics, Inc. All rights reserved.
The OpenEdge® Application Development Environment (ADE) consists of the tools you need to create an OpenEdge application.

This chapter contains the following sections:

- Starting the ADE tools
- Accessing menus and menu options
- Navigating in dialog boxes and windows
- Using the Tools menu
- Using the Help menu
Starting the ADE tools

The ADE comprises the following tools:

- Procedure Editor
- Data Dictionary
- Application Compiler

This manual covers the tools used for application development: Procedure Editor and Application Compiler. For information on the Data Dictionary, refer to *OpenEdge Development: Basic Database Tools*.

**Procedure Editor**

The Procedure Editor lets you create and run OpenEdge procedures. See Chapter 2, “Procedure Editor Tasks” for a description of the tasks you can perform with the Procedure Editor. See Chapter 4, “Procedure Editor Reference,” for a description of the Procedure Editor menu options and dialog boxes.

Before starting the Procedure Editor, be sure to include the directory where you installed OpenEdge in the PATH environment variable (DLC\bin, by default).

There are four ways to start the Procedure Editor:

- To start a multi-user OpenEdge session, enter the following command at the command line, where *database* refers to any database created with OpenEdge Release 10:

  ```
  mpro [ database ]
  ```

- To start a single-user OpenEdge session, enter the following command at the command line, where *database* refers to any database created with OpenEdge Release 10:

  ```
  pro [ database ]
  ```

- To start a single-user OpenEdge session and load procedure files, enter the following command at the command line:

  ```
  pro -param
  " procedure1.p , procedure2.p , procedure3.p , ... "
  ```

- From any of the other ADE tools, choose **Tools**→**Procedure Editor**.
Application Compiler

The Application Compiler lets you compile a group of source procedures (.p and .w files). See Chapter 5, “Application Compiler” for a complete description of how to compile an ABL procedure.

To start the Application Compiler, choose **Tools → Application Compiler** from the menu bar in any of the other OpenEdge tools.
To access and execute the menu options in the ADE tools:

1. Press F3 to access the menu bar for all tools except the Data Dictionary.

2. To choose a menu, do either of the following:
   - Type the mnemonic (the underlined letter) in the menu name. For example, type e to choose the Edit menu.
   - Use the arrow keys to highlight the menu name, then press RETURN.

Some menu options have predefined accelerator keys assigned to them, which let you use the keyboard to choose a menu option. Accelerator keys are function and special key combinations that let you use the keyboard to choose a menu option. If accelerator keys are assigned to an option, they appear to the right of the menu option on pull-down menus.

In some cases, key combinations will not work if you use the ESCAPE key while the CAPS LOCK key is on.

For information about specifying keyboard mappings on your system, see OpenEdge Deployment: Managing ABL Applications.
Navigating in dialog boxes and windows

Many options you choose in the OpenEdge tools display dialog boxes and windows where you have to enter information, toggle on boxes, or choose buttons. Use the following information as a guide to choosing options and entering information into dialog boxes and windows when using OpenEdge tools:

- Move the highlight bar through the dialog box or window by pressing the right arrow or TAB key. For example, press TAB to skip a fill-in area, toggle box, or button.
- Reverse direction and move backward by pressing BACKTAB.
- Scroll up and down lists using the arrow keys.
- Choose a highlighted button or a toggle box by pressing RETURN.
Using the Tools menu

The Tools menu lets you access a tool from another tool. For example, you can access the Data Dictionary from the Procedure Editor. Access to other tools, however, is allowed only in a forward direction. For example, you cannot return to the Procedure Editor from the Data Dictionary without closing the Dictionary. Table 1–1 lists the menu options that are available when you choose the Tools menu option.

Table 1–1: Tools menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure Editor</td>
<td>Accesses the Procedure Editor</td>
</tr>
<tr>
<td>Data Dictionary</td>
<td>Accesses the Data Dictionary</td>
</tr>
<tr>
<td>OS Shell</td>
<td>Escapes to the operating system</td>
</tr>
<tr>
<td>Application Compiler</td>
<td>Accesses the Application Compiler</td>
</tr>
</tbody>
</table>

Tools>Procedure Editor

Choose this option to access the Procedure Editor. The Procedure Editor lets you create, edit, compile, and run ABL procedures. See Chapter 2, “Procedure Editor Tasks” for a complete description of the tasks you can perform with the Procedure Editor.

Tools>Data Dictionary

Choose this option to access the Data Dictionary, which lets you create and modify database schema information including table, field, sequence, indexes, and trigger definitions. You also can generate schema reports. See OpenEdge Development: Basic Database Tools for a complete description of the tasks you can perform with the Data Dictionary.

Tools>OS Shell

Choose this option to temporarily leave the Procedure Editor so you can run operating system commands from your operating system prompt. When you leave the shell, you automatically return to the Procedure Editor.

Tools>Application Compiler

Choose this option to access the Application Compiler. The Application Compiler lets you compile a group of source procedures (.p files). See Chapter 5, “Application Compiler” for a complete description of the Application Compiler.
Using the Help menu

All ADE tools except the Data Dictionary contain the Help menu. Help menu options provide access to online help information about the tool for which you are requesting help. Table 1–2 describes the menu that appears when you choose this option.

Table 1–2: Help menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages...</td>
<td>Displays OpenEdge messages</td>
</tr>
<tr>
<td>Recent Messages...</td>
<td>Displays the most recent OpenEdge message and a detailed description of the message</td>
</tr>
<tr>
<td>Keyboard...</td>
<td>Accesses information about your keyboard and keystrokes</td>
</tr>
<tr>
<td>About Tool...</td>
<td>Shows information about this installation of the Tool</td>
</tr>
</tbody>
</table>

Help>Messages

Choose this option to display a description of OpenEdge messages. Enter a message number to display the description.

Help>Recent Messages

Choose this option to display a description of the most recent message and all other messages generated by OpenEdge during the current session.

Help>Keyboard

Choose this option to show information about Editor and run-time key bindings. These lists show the actions associated with various keystrokes you can perform with your keyboard.

In some cases, key combinations will not work if you use the ESCAPE key while the CAPS LOCK key is on.

Help>About Tool

Choose this option to display the OpenEdge version and copyright date.
Procedure Editor Tasks

This chapter describes many of the tasks you can perform using the Procedure Editor, including:

- Starting the Procedure Editor
- Using edit buffers
- Working with procedures
- Editing text
- Compiling, running, and checking procedures
- Exiting the Procedure Editor

For details on the menu options and dialog boxes of the Procedure Editor, see Chapter 4, “Procedure Editor Reference.”
Starting the Procedure Editor

There are four ways to start the Procedure Editor:

- From any of the other ADE tools, choose Tools→Procedure Editor.

- To start a multi-user OpenEdge session with the Procedure Editor, enter the following command at the command line, where database refers to any database created with OpenEdge Release 10:

  \[ \text{mpro \ [ \text{database} \] } \]

- To start a single-user OpenEdge session with the Procedure Editor, enter the following command at the command line, where database refers to any database created with OpenEdge Release 10:

  \[ \text{pro \ [ \text{database} \] } \]

- To start a single-user OpenEdge session and load procedure files, enter the following command at the command line:

  \[ \text{pro -param " procedure1.p, procedure2.p, procedure3.p, \ldots" } \]

When you start the Procedure Editor, the window shown in Figure 2–1 appears.

![Figure 2–1: Procedure Editor window](image-url)
Table 2–1 describes the features of the Procedure Editor window.

Table 2–1: Procedure editor features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu bar</td>
<td>Lets you access editor commands.</td>
</tr>
<tr>
<td>Insertion point</td>
<td>Marks where text appears when you start typing.</td>
</tr>
<tr>
<td>Procedure area</td>
<td>The visible part of the current buffer where you type and edit your ABL procedures.</td>
</tr>
<tr>
<td>Current buffer name</td>
<td>The name of the procedure file you are currently editing. If the buffer has no name assigned, it appears as “Untitled” and is followed by a number to make it unique.</td>
</tr>
<tr>
<td>Command keys</td>
<td>Allow you to perform basic tasks in the Procedure Editor. These keys let you run procedures and access menus and help. They also let you open files into buffers and save or close them.</td>
</tr>
</tbody>
</table>

Dialog boxes, windows, or frames might appear in front of the window for various reasons. For example, they can display alert boxes that enable you to enter search strings and filenames or verify actions.
Using edit buffers

An *edit buffer* is a work area where you write and edit an ABL procedure. The Procedure Editor creates an edit buffer for each procedure you are working on, allowing you to open and edit multiple procedures in a single session. The number of buffers is limited by system memory.

Although you can open multiple buffers in the Procedure Editor, you can open only one buffer for each operating system file.

The buffer displayed in the Procedure Editor’s procedure area is called the *current buffer*. When you enter text into the procedure area of the Procedure Editor, you enter it into the current buffer. When you start the Procedure Editor, the current buffer is empty and untitled unless you specify a procedure at the command line.

The maximum buffer size you can open in the Procedure Editor is the lesser of the following:

- \(32,767 \times \text{bytesPerLine}\), where *bytesPerLine* is the width of your terminal
- Available system resources
Listing open buffers

The Procedure Editor creates buffers when you use the **New** or **Open** option. To view a list of the open buffers for your current session, choose **Buffer → List**. The open buffers are listed in the **Buffer List** dialog box in the order that you opened them, as shown in Figure 2–2. For more information on the **Buffer List** dialog box, see Chapter 4, “Procedure Editor Reference.”

![Buffer List dialog box](image)

**Figure 2–2:** **Buffer List** dialog box

Switching buffers

To switch to another open buffer, use one of the following techniques:

- Choose **Buffer → Next Buffer**. The Procedure Editor makes the next buffer in the buffer list the current buffer.

- Choose **Buffer → Previous Buffer**. The Procedure Editor makes the previous buffer in the buffer list the current buffer.

- Choose **Buffer → List**, then select a buffer from the **Buffer List** dialog box and choose **OK**. The Procedure Editor displays the buffer you selected as the current buffer.

Displaying buffer information

To display information about the current buffer, choose **Buffer → Information**. OpenEdge displays the **Buffer Information** dialog box, which contains the filename; read and write access for the file; the number of lines, columns, and bytes; and the modified status of the buffer. Choose **OK** to close the dialog box. See Chapter 4, “Procedure Editor Reference” for more information on the **Buffer Information** dialog box.
Working with procedures

This section describes how to manipulate procedures in the Procedure Editor. It describes the following tasks:

- Creating a new procedure
- Opening a procedure
- Saving a procedure
- Printing a procedure
- Closing a procedure

Creating a new procedure

To create a new procedure, choose File→New from the menu bar. OpenEdge creates a new buffer in the Procedure Editor and makes the new buffer the current buffer. The Procedure Editor adds the new buffer to the buffer list, labels it “Untitled,” and assigns it a number. To use this new buffer to write an ABL procedure, enter code and choose the Save As option to name and store the procedure.

Note: The Procedure Editor does not allow you to create a file with a comma in its name.

Opening a procedure

To open an existing procedure file:

1. Choose File→Open.
2. Select the filename from the Open dialog box, then choose OK.

The Procedure Editor cannot open a file with a comma in its name.

The Procedure Editor creates a new buffer, reads the file into it, makes it the current buffer, and labels it with the name of the file. You can open multiple buffers in the Procedure Editor, but you cannot open more than one buffer at a time for the same operating system file. The number of buffers is limited by system memory. You can access other buffers from the Buffer menu. For more information on the Buffer menu, see Chapter 4, “Procedure Editor Reference.”

The maximum buffer size you can open in the Procedure Editor is the lesser of the following:

- $32,767 \times \text{bytesPerLine}$, where bytesPerLine is the width of your terminal
- Available system resources
Saving a procedure

You can save the contents of the current buffer by:

- Saving a procedure to a new file
- Saving a procedure to the current file

Saving a procedure to a new file

To save a new, untitled procedure to a file:

1. From the menu bar, choose File→Save or File→Save As. The Save As dialog box appears. See Chapter 4, “Procedure Editor Reference” for a detailed description of the Save As dialog box fields:

   ![Save As Dialog Box](image)

   - **File Name:**
     - `/tmp_mnt/users/doc/spurr`
   - **Files:**
     - proc1.p
     - proc2.p
     - proc3.p
     - proc4.p
     - proc5.p
     - proc6.p
   - **Directories:**
     - desktop
     - help
     - Inst5CC
     - local
   - **<OK>**
   - **<Cancel>**

2. Specify the file where you want to save the procedure.

   When you enter a filename, add the `.p` extension to the procedure file. Procedures are stored as operating system files. UNIX filenames are case-sensitive, unlike Windows filenames. Unless you specify a path when you name the file, OpenEdge stores the file in your current working directory.

   **Caution:** Do not name a procedure `_edit.p`, because that is the name of the Procedure Editor. If you do, and the procedure falls before `$DLC` in your `PROPATH`, OpenEdge accesses the incorrect file when you try to run the Procedure Editor.

OpenEdge saves the procedure to the specified filename and keeps the buffer open so you can continue working in it. The buffer name changes to the specified filename.

The Procedure Editor cannot save a file with a comma in its name.

Saving a procedure to the current file

When you are in a named buffer and you choose File→Save, OpenEdge automatically saves the procedure to the current filename and keeps the buffer open so you can keep working in it.
Printing a procedure

To print the contents of the current buffer, choose File→Print. OpenEdge sends the current buffer contents to your computer system’s default printer. There are no print format options.

Closing a procedure

To close a procedure, choose File→Close. The Procedure Editor closes the current buffer. If you make changes to the current buffer, OpenEdge prompts you to save the changes before it deletes the buffer. If the buffer is untitled and you choose to save it, OpenEdge displays the Save As dialog box. Specify the filename and choose OK to save the procedure.
Editing text

This section describes how to edit text in the Procedure Editor. It describes the following tasks:

- Entering text
- Selecting text
- Cutting, copying, and pasting text
- Searching for text
- Inserting a file into a procedure
- Inserting a field name into a procedure

Entering text

Enter text into the Procedure Editor by typing as you would in any other editor. Text you enter into the current buffer appears at the location of the cursor. The cursor is a place marker in text. The location of the cursor is referred to as the insertion point.

You can use the keyboard arrow keys to relocate the insertion point. You also can choose Search→Goto Line to move the insertion point to a specific line number within the current buffer. See the “Search>Goto Line” section on page 4–11 for more information on this option.

Entering special characters

There are some characters that you cannot type directly into the edit buffer. To represent ASCII control characters or other character codes that the keyboard cannot generate directly (8-bit codes, for example), type the three-digit octal code of the character, preceded by a tilde (~). For more information about special characters, see OpenEdge Development: ABL Reference.

When you write procedures with the Procedure Editor, you can use uppercase, lowercase, or mixed case. The ABL Compiler recognizes “table,” “TABLE,” and “Table” as the same word.

Entering long text lines

You can create and edit text lines longer than 80 characters using the Procedure Editor.

You can use a tilde (~) as the last non-blank character on a line to indicate that the line is continued, beginning with column 1 of the next line in the window, as shown in the following example:

```
DISPLAY "This is a long message ~
continued on the next line".
```

The ABL Compiler interprets the lines in this example as the following single line:

```
DISPLAY "This is a long message continued on the next line".
```

The Procedure Editor does not change the physical presentation you give the file. When you use a tilde to connect two lines, the Procedure Editor maintains the tilde in the code. The Procedure Editor saves it and displays it; however, the Compiler compiles the file without the tilde.
Selecting text

Select text when you want to perform an action or command on that text or if you want to use the selected text in a command.

You can select text by positioning the cursor where you want to begin selecting text and pressing \texttt{CTRL+V} to enter block selection mode. Then move the cursor to the end of the text you want to select. The text is not highlighted, but it is selected.

Cutting, copying, and pasting text

To cut or copy text in a procedure:

1. Select the text you want to cut or copy. See the “Selecting text” section above for the steps to follow.

2. To cut the text, choose \texttt{Edit} → \texttt{Cut}. To copy the text, choose \texttt{Edit} → \texttt{Copy}.

   The Procedure Editor deletes or copies the selected range of text from the file in the current buffer and places it onto the clipboard.

To paste previously copied text into a procedure:

1. Position the cursor at the point you want to insert the text.

2. Choose \texttt{Edit} → \texttt{Paste}.

   The Procedure Editor inserts the contents of the clipboard at the cursor and repositions the clipboard after the pasted text.

Searching for text

You can search for text in the current buffer using the following menu options:

- Finding text
- Finding the next or previous occurrence of text
- Replacing text

Finding text

To search for text in the current buffer:

1. Choose \texttt{Search} → \texttt{Find}. The \texttt{Find} dialog box appears:

   \begin{center}
   \begin{tabular}{|l|}
   \hline
   \texttt{Find What:} [ ] \hline
   \texttt{Match Case} \quad \texttt{Direction:} \quad \texttt{Wrap at Beginning/End} \quad \texttt{Up} \quad \texttt{Down} \hline
   \texttt{OK} \quad \texttt{Cancel} \hline
   \end{tabular}
   \end{center}
2. Enter the text string you want to find.

   If you have searched for a string in the current session, the **Find What** field displays the text string for which you last searched. If your search is not case sensitive (that is, you do not select the **Match Case** option in the **Find** dialog box), the string you supply can be uppercase, lowercase, or both; otherwise, enter it exactly as you want to search for it.

3. Specify whether the search is case sensitive using the **Match Case** option.

4. Specify whether to wrap to the beginning or end of the current buffer when the search reaches the opposite end of the buffer.

5. Specify whether to search forward or backward through the current buffer by selecting the up or down options.

6. Choose **OK**.

   The Procedure Editor searches for the first occurrence of the string in the direction you specify. When it finds a match for the search string, it positions the cursor at the end of the search string. You can choose **Search**→**Find Next** or **Search**→**Find Previous** to find the next or previous occurrence of the text. The Procedure Editor displays an alert box if it does not find a match.

### Finding the next or previous occurrence of text

If you have used the **Search**→**Find** option in the current session, you can search for text using the **Search**→**Find Next** or **Search**→**Find Previous** menu options. The Procedure Editor searches for the text you specified in the **Find What** field of the **Find** dialog box in the forward direction (**Find Next**) or reverse direction (**Find Previous**).

If you have not previously used the **Search**→**Find** option in this session, OpenEdge displays an alert box when you select **Search**→**Find Next** or **Search**→**Find Previous**.

### Replacing text

To search for and replace text in the current buffer:

1. Choose **Search**→**Replace**.

2. Enter the text string you want to find. The **Replace** dialog box appears:

   ![Replace dialog box](image)

   If you have searched for a string in the current session, the **Find What** field displays the string for which you last searched. If your search is not case sensitive (that is, you do not select the **Match Case** option in the **Replace** or **Find** dialog box), the string you supply can be uppercase, lowercase, or both; otherwise, enter it exactly as you want to search for it. You cannot use wildcard characters when specifying the find string.

3. Enter the string with which you want to replace the specified string.

4. Specify if the search is case sensitive.
5. Choose the **Replace All** button if you want to replace all occurrences of the search string with the new string without confirming each occurrence.

6. Choose **OK**.

The Procedure Editor searches for the first occurrence of the string in a forward direction. If you do not choose the **Replace All** button, the Editor prompts you to confirm the replacement for each occurrence of the string it finds. If you choose the **Replace All** button, the Editor automatically replaces each occurrence of the string with the text in the **Replace With** field. When complete, the Editor displays an information alert box indicating the **Replace All** task is complete and displaying the number of occurrences replaced.

### Inserting a file into a procedure

To insert the entire contents of a file into the text in the current buffer:

1. Place the cursor at the point where you want to insert the file.
2. Choose **Edit** → **Insert File** from the main menu. The **Insert File** dialog box appears.
3. Choose the file you want to insert, then choose **OK**.

The Procedure Editor inserts the file contents into the current buffer at the cursor position.

### Inserting a field name into a procedure

To insert a field name into a procedure in the current buffer:

1. Place the cursor at the point where you want to insert the field name.
2. Choose **Edit** → **Insert Fields** from the main menu. If you are connected to a database, The **Field Selector** dialog box appears:

   ![Field Selector](image)

   **Note:** If you are not already connected to a database, OpenEdge displays an alert box. If you choose **OK**, OpenEdge displays the **Database Connect** dialog box. See *OpenEdge Development: Basic Database Tools* for information on this dialog box. After you connect to a database, the **Field Selector** dialog box appears.

3. Choose the fields you want to insert.
4. Specify prefixes to include with the field names. See the “Edit>Insert Fields” section on page 4–8 for more information on these options.

5. Choose OK. The Procedure Editor inserts the field names into the current buffer at the cursor position.
Compiling, running, and checking procedures

The Procedure Editor supports the full edit-compile-run cycle. This support includes:

- Checking a procedure’s syntax
- Compiling and running a procedure
- Debugging a procedure

Note: OpenEdge provides startup options that let you see compile-time alerts if the source code contains specified ABL statements, or that let you disable specified ABL keywords. For more information, see the “Startup parameters for Compiler behavior” section on page 5–2.

Checking a procedure’s syntax

To check a procedure’s syntax, choose **Compile → Check Syntax** from the menu bar. The Procedure Editor checks the procedure’s syntax and displays all applicable messages in the **Compiler Messages** dialog box.

Compiling and running a procedure

To compile and run a procedure in the current buffer, choose **Compile → Run** from the menu bar. The Procedure Editor accesses the ABL Compiler, which compiles the procedure. If the procedure does not compile, OpenEdge displays all applicable messages in the **Compiler Messages** dialog box.

The messages displayed typically show a brief description of the message, often including the name of the file and line number containing the error, followed by an OpenEdge message number in parentheses. You can display additional information about the messages using the **Help → Messages** or **Help → Recent Messages** options.

If the Compiler detects any errors, the Procedure Editor moves the text cursor to the line in the current buffer that contains the first error. If the file containing the error is not in the current buffer but is an include file, the Procedure Editor opens the include file, making it the current buffer. The Procedure Editor then positions the cursor on the line in the include file that contains the error. If the include file containing the error is not syntactically complete (that is, it does not contain at least one ABL statement with a period), the Procedure Editor opens the source file that references the include statement, rather than the include file that contains the error.

When you run a procedure and have several open buffers, the Compiler accesses only the code in the current buffer of the Procedure Editor. It does not access files in other buffers but uses the saved versions of the files. This means that if you run a procedure, and the Compiler detects an error in a file open in a buffer other than the current buffer, you must save any changes you make to that called file for them to be recognized when you rerun the initial procedure.
If you do not save the changes, the next time you run the initial procedure that calls the file, the Compiler runs the saved version of the files and disregards the changes you have made in the open buffer of the Procedure Editor. When the Compiler detects the error, the Procedure Editor switches the called file to the current buffer and places the cursor on the line that previously contained the error. Your changes appear in the current buffer, but because you did not save them, the Compiler could not access the modified code.

**Debugging a procedure**

To debug a procedure, choose **Compile → Debug** from the menu bar. The Procedure Editor checks the procedure’s syntax. If the procedure compiles, the Procedure Editor opens the OpenEdge Debugger and displays the procedure at the break point of the first executing line. For more information about debugging a procedure, see *OpenEdge Development: Debugging and Troubleshooting.*
Exiting the Procedure Editor

You can exit the Procedure Editor in two ways:

- Choose File→ Exit.
- Type quit in an empty buffer, then choose Compile→ Run or press GO.

If there are open buffers with unsaved changes, an alert box prompts you to save or discard the buffers before leaving the Procedure Editor. To discard the buffers and exit the Procedure Editor, choose No in the alert box.

To save any changed buffers:

1. Choose Yes in the alert box. OpenEdge displays the Save Buffers with Changes dialog box.

2. Select the buffers to save and choose the Save Selected button. If you are saving any untitled buffers, the Save As dialog box appears for each buffer.

3. Specify the filename for each untitled buffer, then choose OK. After you specify the filename for the last untitled buffer, OpenEdge closes the Procedure Editor.

When OpenEdge closes the Procedure Editor, it returns to where you started. For example, if you start the Procedure Editor from the operating system command line, OpenEdge returns to the command line.
This chapter explains how to add integration hooks to the Procedure Editor by describing the parameters and events for `adecomm/_adeevnt.p` and how to modify this procedure to interface with source code management or your own proprietary tools:

- **ADE event** (`adecomm/_adeevnt.p`)
- **Parameters**
- **Events**
- **Usage**

Source code management tools or your own proprietary tools can interface with the Procedure Editor. They can intercept and augment tool behavior at the following critical points in the application development process:

- Opening, closing, and saving files
- Before and after a file is run, debugged, or checked for syntax
- Startup and shutdown of the Procedure Editor

To allow source code management or your own proprietary tools to “trap” these ADE events, the Procedure Editor calls a procedure file, `adecomm/_adeevnt.p`, at those critical processing points. You can modify this procedure to intercept and augment standard tool behavior. The source code to this procedure is provided by Progress Software Corporation. To use this procedure, you must copy the source code, modify it, compile it, and place it in your PROPATH.

**Note:** For GUI application development, there are three procedure files for integration hooks—`adecomm/_adeevnt.p`, `adecomm/_getfile.p`, and `adecomm/_chosobj.w`. For more information on how to use integration hooks for GUI application development, see the on-line help.
You can modify adecomm/_adeevnt.p to intercept or filter various events that occur in an ADE session. The Procedure Editor calls adecomm/_adeevnt.p when certain processing events occur. Generally, these events are file oriented (for example, OPEN, SAVE, and CLOSE), but they can be more general (for example, STARTUP and SHUTDOWN).

The source code for this file is in DLC/src/adecomm/_adeevent.p. Look at this file for the latest information on ADE events.
Parameters

This section describes the input and output parameters for the adecomm/_adeevnt.p procedure.

Input parameters

The adecomm/_adeevnt.p procedure takes the following input parameters:

- **p_product**
  The ADE product code (for the Procedure Editor, use Editor).

- **p_event**
  The name of an event: NEW, OPEN, BEFORE-OPEN, CLOSE, BEFORE-CLOSE, SAVE, BEFORE-SAVE, COMPILE, BEFORE-COMPILE, RUN, BEFORE-RUN, DEBUG, BEFORE-DEBUG, CHECK-SYNTAX, BEFORE-CHECK-SYNTAX, CHECK-SYNTAX-PARTIAL, and BEFORE-CHECK-SYNTAX-PARTIAL.

- **p_context**
  A string used to uniquely identify the file being edited. The Procedure Editor uses the edit widget-handle of the buffer for this value. A specific file has the same context ID for all its file operations; however, if you close a file and then open it again, the context number will change.

- **p_other**
  Additional information passed about an event (for example, a SAVE event normally passes the filename for the save).

  The current filename associated with the window. The name is unknown (?) if it was not set; for example, after a **File→New**.

Output parameters

The adecomm/_adeevnt.p procedure uses the following output parameters:

- **p_ok**
  A logical value used to approve or cancel a subset of ADE events. For example, returning a FALSE value from the BEFORE-SAVE event cancels the save.
Events

This section describes the file-specific events and the STARTUP and SHUTDOWN events for the adec/adevnt.p procedure.

File-specific events

Table 3–1 describes the events (corresponding to p_event) that are related to file operations.

Table 3–1: File-specific events

<table>
<thead>
<tr>
<th>Event</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE-CHECK-SYNTAX</td>
<td>Called before a Check Syntax. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-CHECK-SYNTAX-PARTIAL</td>
<td>Called before a Partial Check Syntax. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-CLOSE</td>
<td>Called before a file is to be closed. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-COMPILE</td>
<td>Called before a file is to be compiled. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-DEBUG</td>
<td>Same as BEFORE-RUN, except DEBUG is chosen.</td>
</tr>
<tr>
<td>BEFORE-OPEN</td>
<td>Called before a file is to be opened. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-RUN</td>
<td>Called before a file is written to disk for a run. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>BEFORE-SAVE</td>
<td>FALSE before a file is to be saved. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>CHECK-SYNTAX</td>
<td>Called after a Check Syntax.</td>
</tr>
<tr>
<td>CHECK-SYNTAX-PARTIAL</td>
<td>Called after a Partial Check Syntax.</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Called after a window or buffer is closed.</td>
</tr>
<tr>
<td>COMPILE</td>
<td>Called after a file is compiled.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Same as RUN, except DEBUG was chosen. Returning p_ok as FALSE cancels the operation.</td>
</tr>
<tr>
<td>NEW</td>
<td>Called after a new window/dialog box is created.</td>
</tr>
<tr>
<td>OPEN</td>
<td>Called after a file is opened.</td>
</tr>
<tr>
<td>RUN</td>
<td>Called after RUN of file ends.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Called after a file is saved.</td>
</tr>
</tbody>
</table>

Some events—for example NEW—cannot be cancelled even if returning FALSE.
STARTUP and SHUTDOWN events

The following Procedure Editor events (corresponding to p_event) occur at Procedure Editor startup and shutdown:

- **STARTUP** — Called when the Procedure Editor (PE) has been loaded and initialized. This call occurs immediately before user input is allowed. In this case:
  
  - `p_context = STRING(procedure-handle-of-the-PE-main-routine)`
  - `p_other = STRING(widget-handle-of-the-PE-window)`

- **SHUTDOWN** — Called when a user requests that the Procedure Editor shutdown. This call occurs before any settings have been saved or items destroyed. In this case:
  
  - `p_context = STRING(procedure-handle-of-the-PE-main-routine)`
  - `p_other = STRING(widget-handle-of-the-PE-window)`

The Procedure Editor STARTUP and SHUTDOWN events cannot be cancelled.
Usage

The following comments address Procedure Editor usage issues:

- **BEFORE-CLOSE and CLOSE** — Technically, `p_other` should be unknown (?) after a file closes; however, this parameter still shows the last available file name for the procedure file. Unknown (?) appears only if there is no filename.

- **BEFORE-RUN, RUN, BEFORE-DEBUG, DEBUG, BEFORE-CHECK-SYNTAX, and CHECK-SYNTAX** — These all use the last specified filename as `p_other`. The actual file being run or checked is a temporary file with a name like `p01928384.ped`. This name is not used.

- If the user tries to close a buffer, then the Procedure Editor first prompts the user to save. The entire save operation events fire before the call to BEFORE-CLOSE.

- **NEW** is called after a buffer is created. You see the buffer before the **NEW** event is called. All events are called after the event has finished.

- When you control the handle for the Procedure Editor window, you can manipulate the display of the window. For example, you can add a menu option.

The scenario shown in Table 3–2 applies to the Procedure Editor.

**Table 3–2: Usage scenario for the Procedure Editor**

<table>
<thead>
<tr>
<th>Action</th>
<th>p_event</th>
<th>p_context</th>
<th>p_other</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Creating, saving, and running the file</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File→ New</td>
<td>NEW</td>
<td>56788</td>
<td>?</td>
<td>Filename unknown.</td>
</tr>
<tr>
<td>File→ Save</td>
<td>BEFORE-SAVE</td>
<td>56788</td>
<td>c:\9\window-1.w</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>SAVE</td>
<td>56788</td>
<td>c:\9\window-1.w</td>
<td>–</td>
</tr>
<tr>
<td>Running the file</td>
<td>BEFORE-RUN</td>
<td>56788</td>
<td>c:\9\window-1.w</td>
<td>Run uses last filename.</td>
</tr>
<tr>
<td></td>
<td>RUN</td>
<td>56788</td>
<td>c:\9\window-1.w</td>
<td>–</td>
</tr>
<tr>
<td>File→ Save As</td>
<td>BEFORE-SAVE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>Same context, new name.</td>
</tr>
<tr>
<td></td>
<td>SAVE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>–</td>
</tr>
<tr>
<td><strong>Closing the file with modifications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure Editor</td>
<td>BEFORE-SAVE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>Closing a file can cause the Procedure Editor to prompt the user to save a file.</td>
</tr>
<tr>
<td></td>
<td>SAVE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>–</td>
</tr>
<tr>
<td>Procedure Editor</td>
<td>BEFORE-CLOSE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>CLOSE</td>
<td>56788</td>
<td>c:\9\my-file.w</td>
<td>–</td>
</tr>
<tr>
<td>Action</td>
<td>p_event</td>
<td>p_context</td>
<td>p_other</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>--------------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Opening the file</td>
<td>BEFORE-OPEN</td>
<td>?</td>
<td>c:\9\my-file.w</td>
<td>Occurs after the Open dialog box. Before-open can be cancelled.</td>
</tr>
<tr>
<td></td>
<td>OPEN</td>
<td>46647</td>
<td>c:\9\my-file.w</td>
<td>Observe the new context number.</td>
</tr>
<tr>
<td>Closing the file without modifications</td>
<td>BEFORE-CLOSE</td>
<td>46647</td>
<td>c:\9\my-file.w</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>CLOSE</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>
This chapter describes the Procedure Editor’s menu options and dialog boxes.

The Procedure Editor is a menu-driven tool you can use to create, edit, compile, and run ABL procedures. The Procedure Editor is fully integrated with the ABL Compiler, the OpenEdge Data Dictionary, and the OpenEdge Debugger. It checks the syntax of your procedures and verifies the existence of tables and fields named in your ABL procedures.

This chapter contains the following sections:

- Procedure Editor menu bar
- File menu
- Edit menu
- Search menu
- Buffer menu
- Compile menu
- Tools menu
- Help menu

For information on how to use the Procedure Editor to perform tasks, see Chapter 2, “Procedure Editor Tasks.”
Procedure Editor menu bar

The following sections describe the menus and options available from the Procedure Editor. Figure 4–1 shows the Procedure Editor menu bar. To access the menu bar, press F3.

![Figure 4–1: Procedure Editor menu bar](image)

Table 4–1 describes the Procedure Editor menus.

<table>
<thead>
<tr>
<th>Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Manages files and exits the Procedure Editor</td>
</tr>
<tr>
<td>Edit</td>
<td>Manipulates and edits blocks of code</td>
</tr>
<tr>
<td>Search</td>
<td>Searches for and replaces code strings in the current buffer</td>
</tr>
<tr>
<td>Buffer</td>
<td>Manages buffers, including switching the current buffer and displaying buffer information</td>
</tr>
<tr>
<td>Compile</td>
<td>Runs and compiles ABL procedure files</td>
</tr>
<tr>
<td>Tools</td>
<td>Accesses other OpenEdge tools</td>
</tr>
<tr>
<td>Help</td>
<td>Accesses information about OpenEdge error messages and keyboard mappings</td>
</tr>
</tbody>
</table>
File menu

The File menu options allow you to retrieve and save ABL program files. Table 4–2 describes the menu that appears when you choose this option.

Table 4–2: File menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Creates a new file or procedure</td>
</tr>
<tr>
<td>Open</td>
<td>Opens an existing file or procedure to edit</td>
</tr>
<tr>
<td>Close</td>
<td>Closes the current buffer</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the contents of the current buffer</td>
</tr>
<tr>
<td>Save As</td>
<td>Saves the contents of the current buffer to a file</td>
</tr>
<tr>
<td>Print</td>
<td>Prints the contents of the current buffer</td>
</tr>
<tr>
<td>Exit</td>
<td>Ends your current Procedure Editor session</td>
</tr>
</tbody>
</table>

File>New

Choose this option to create a new file or procedure. The Procedure Editor creates a new, untitled buffer and makes it the current buffer. The number of buffers is limited by system memory.

The maximum buffer size you can open in the Procedure Editor is the lesser of the following:

- 32,767*bytesPerLine, where bytesPerLine is the width of your terminal
- Available system resources

You can switch between buffers using the Buffer menu commands. For more information, see the “Buffer menu” section on page 4–12.

File>Open

Choose this option to edit an existing file. When you choose this option, the Open dialog box shown in Figure 4–2 appears.

![Open dialog box](image)

Figure 4–2: Open dialog box
The Procedure Editor cannot open a file with a comma in its name.

When you choose the **Files** button from the **Open** dialog box, the **Files** dialog box like the one shown in **Figure 4–3** appears.

![Files dialog box](Image)

**Figure 4–3:** Files dialog box

Table 4–3 describes the user-interface elements of the **Files** dialog box.

<table>
<thead>
<tr>
<th>User-interface element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Specifies the name of the file you want to open.</td>
</tr>
<tr>
<td>Files</td>
<td>Lists the files in the currently selected directory.</td>
</tr>
<tr>
<td>Directories</td>
<td>Shows the currently selected directory. The directory selection list displays all the available directories.</td>
</tr>
</tbody>
</table>

**File>Close**

Choose this option to delete the current buffer. If you make changes to the current buffer, the Procedure Editor prompts you to save the changes to a file before it deletes the buffer. If the buffer is untitled, an alert dialog box like the one in **Figure 4–4** appears.

![Close alert dialog box](Image)

**Figure 4–4:** Close alert dialog box
**File>Save**

Choose this option to save the contents of the current buffer to the current file. When you choose this option and the current buffer is untitled, the **Save As** dialog box is displayed, as shown in **Figure 4–5**. The Procedure Editor displays an alert box if you try to save changes to a read-only file.

**File>Save As**

Choose this option to save the contents of the current buffer to a new file or to an existing file. When you choose this option, the **Save As** dialog box shown in **Figure 4–5** appears.

![Save As dialog box]

**Figure 4–5: Save As dialog box**

**Note:** The Procedure Editor cannot save a file with a comma in its name.

Table 4–4 describes the fields in the **Save As** dialog box.

**Table 4–4: Save As dialog box fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Specifies the name for the file you want to save</td>
</tr>
<tr>
<td>Files</td>
<td>Lists the files in the selected directory</td>
</tr>
<tr>
<td>Directories</td>
<td>Lists the directories in your file system</td>
</tr>
</tbody>
</table>

**File>Print**

Choose this option to print the contents of the current buffer. The current buffer contents are sent to your computer system’s default printer. There are no special formatting features or options.
File>Exit

Choose this option to close the Procedure Editor. When you exit the Procedure Editor, you return to where you started. For example, if you start the Procedure Editor from the operating system command line, you return to the command line.

If there are open buffers with unsaved changes, an alert box prompts you to save or discard the changes before leaving the Procedure Editor. If you choose to save the changes, the Save Buffers with Changes dialog box shown in Figure 4–6 appears. This dialog box lists all the open buffers that have unsaved changes.

![Save Buffers with Changes dialog box](image)

**Figure 4–6: Save Buffers with Changes dialog box**

If you select any untitled buffers and choose the Save Selected button, the Save As dialog box appears for each untitled buffer. You can specify a filename for each untitled buffer or cancel the save operation.
Edit menu

The **Edit** menu options allow you to manipulate and edit blocks of text. Table 4–5 describes the menu that appears when you choose this option.

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Removes selected text from the current buffer and stores it on the system clipboard</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies selected text onto the system clipboard</td>
</tr>
<tr>
<td>Paste</td>
<td>Places clipboard contents in the current buffer at the insertion point</td>
</tr>
<tr>
<td>Insert File</td>
<td>Copies the contents of the selected file into the current buffer at the insertion point</td>
</tr>
<tr>
<td>Insert Fields</td>
<td>Inserts selected database field names into the current buffer at the insertion point</td>
</tr>
</tbody>
</table>

**Edit>Cut**

Choose this option to delete the selected text from the file in the current buffer and place it onto the system clipboard, erasing the previous clipboard contents.

Choose the **Edit→Paste** option to retrieve the deleted text from the clipboard and insert it into the current buffer.

To select text, position the cursor where you want to begin selecting text and press **CTRL+V** to enter block selection mode. Then move the cursor to the end of the text you want to select. The text is not highlighted, but it is selected.

**Edit>Copy**

Choose this option to make a copy of the selected text from the file in the current buffer and place it onto the system clipboard, erasing the previous clipboard contents.

Choose the **Edit→Paste** option to retrieve the copied text from the clipboard and insert it into the current buffer.

To select text, position the cursor where you want to begin selecting text and press **CTRL+V** to enter block selection mode. Then move the cursor to the end of the text you want to select. The text is not highlighted, but it is selected.

**Edit>Paste**

Choose this option to place the contents of the system clipboard at the insertion point in the file in the current buffer. If text is selected, the **Paste** option replaces it with the contents of the system clipboard.
Edit>Insert File

Choose this option to read a copy of a file into the current buffer. When you choose this option, the Insert File dialog box shown in Figure 4–7 appears.

The Edit→Insert File option works the same way as the File→Open option, except that it pulls the file information into the current buffer instead of a new buffer.

![Insert File dialog box](image)

Figure 4–7: Insert File dialog box

Edit>Insert Fields

Choose this option to insert a field name into the current buffer. When you choose this option, the Field Selector dialog box shown in Figure 4–8 appears.

If you are not already connected to a database, OpenEdge displays an alert box and lets you access the Database Connect dialog box.

![Field Selector dialog box](image)

Figure 4–8: Field Selector dialog box

Table 4–6 describes the user-interface elements of the Field Selector dialog box.

Table 4–6: Field Selector dialog box elements

<table>
<thead>
<tr>
<th>User-interface element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Indicates you do not want to include a prefix with the field name in your procedure</td>
</tr>
<tr>
<td>Table</td>
<td>Indicates you want to include the table name as well as the field name in your procedure</td>
</tr>
<tr>
<td>Database.Table</td>
<td>Indicates you want to include the database and table name with the field name in your procedure</td>
</tr>
</tbody>
</table>
Search menu

The **Search** menu options allow you to search for and replace text strings in the current buffer. **Table 4–7** describes the menu that appears when you choose this option.

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
<td>Searches for specific text</td>
</tr>
<tr>
<td>Find Next</td>
<td>Searches for the next occurrence of the text string specified in the <strong>Find</strong> dialog box</td>
</tr>
<tr>
<td>Find Previous</td>
<td>Searches for the previous occurrence of the text string specified in the <strong>Find</strong> dialog box</td>
</tr>
<tr>
<td>Replace...</td>
<td>Searches for and changes specified text</td>
</tr>
<tr>
<td>Goto Line...</td>
<td>Moves the insertion point to a specified line number in the current buffer</td>
</tr>
</tbody>
</table>

**Search>Find**

Choose this option to search for text within a file. When you choose **Search→Find**, the **Find** dialog box shown in **Figure 4–9** appears.

![Find dialog box](image)

**Figure 4–9: **Find dialog box**

**Table 4–8** describes the fields in the **Find** dialog box.

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find What</td>
<td>Identifies the text to search for.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Specifies whether the search is case-sensitive.</td>
</tr>
<tr>
<td>Wrap at Beginning/End</td>
<td>Specifies whether to wrap to the beginning of the current buffer and continue searching when the search reaches the end of the buffer.</td>
</tr>
<tr>
<td>Direction</td>
<td>Specifies whether to search forward or backward through the current buffer. Choose the appropriate radio button to indicate the search direction.</td>
</tr>
</tbody>
</table>
**Search>Find Next**

Choose this option to find the next occurrence of the text for which you most recently searched using the Search→Find option. This option uses the same search criteria as the Search→Find option.

**Search>Find Previous**

Choose this option to find the previous occurrence of the text for which you most recently searched using the Search→Find option. This option uses the same search criteria as the Search→Find option.

**Search>Replace**

Choose this option to search for and replace text within the current buffer. This type of search is always in a forward direction.

When you choose this option, the Replace dialog box shown in Figure 4–10 appears.

![Replace dialog box](image)

**Figure 4–10:  Replace dialog box**

Table 4–9 describes the user-interface elements of the Replace dialog box.

<table>
<thead>
<tr>
<th>User-interface element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find What</td>
<td>Identifies the text to search for</td>
</tr>
<tr>
<td>Replace With</td>
<td>Specifies the new text you want to insert</td>
</tr>
<tr>
<td>Match Case</td>
<td>Specifies whether the search is case-sensitive</td>
</tr>
</tbody>
</table>

The Find What and Replace With fill-in fields default to the text and options you entered the last time you executed the option in the current session.
Search>Goto Line

Choose this option to move the insertion point to a specific line number within the current buffer.

When you choose this option, the Goto Line dialog box appears as shown in Figure 4–11. The Goto Line command lets you enter the line number. It defaults to the line number where the cursor is located in the current buffer. If you enter a number that exceeds the number of lines in the current buffer, OpenEdge moves the cursor to the end of the buffer.

![Goto Line dialog box](image)

Figure 4–11: Goto Line dialog box
Buffer menu

The Buffer menu options allow you select and view multiple open buffers. Each time you choose File→New or File→Open, the Editor creates a buffer containing a copy of a file. Table 4–10 describes the menu that appears when you choose this option.

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List</td>
<td>Lists the open buffers</td>
</tr>
<tr>
<td>Next Buffer</td>
<td>Displays the next open buffer</td>
</tr>
<tr>
<td>Previous Buffer</td>
<td>Displays the previous open buffer</td>
</tr>
<tr>
<td>Information</td>
<td>Views information about the current buffer</td>
</tr>
</tbody>
</table>

Buffer>List

Choose this option to display a list of open buffers. Buffers appear listed in the order in which you open them. When you choose this option, the Buffer List dialog box shown in Figure 4–12 appears.

![Buffer List dialog box](image)

This list has a marker (>) that indicates the current buffer. You can switch to another buffer by selecting it from the list and pressing GO.

An asterisk (*) marks each buffer modified since it was last saved.

Buffer>Next Buffer

Choose this option to display the next buffer in the buffer list. When you choose this option, the next open buffer appears and becomes the current buffer. Buffers appear in the order in which you open them. If you choose Next when the last buffer in the list is the current buffer, the first buffer in the list appears.
**Buffer>Previous Buffer**

Choose this option to display the previous buffer in the buffer list. When you choose this option, the previous open buffer appears and becomes the current buffer. Buffers appear in the order in which you open them. If you choose **Previous** when the first buffer in the list is the current buffer, the last buffer in the list appears.

**Buffer>Information**

Choose this option to view settings for the current buffer. When you choose this option, a **Buffer Information** dialog box like the one in Figure 4–13 appears.

![Buffer Information dialog box](Figure 4–13)

Table 4–11 describes the user-interface elements of the **Buffer Information** dialog box.

<table>
<thead>
<tr>
<th>User-interface element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Displays the name of the operating system file from which the buffer was read. It always displays the full pathname of the file.</td>
</tr>
<tr>
<td>File Access</td>
<td>Specifies whether the file from which the text in the current buffer was read is read-only or updatable. If the file is read-only, you cannot save changes to the current buffer in the original file. You must choose <strong>File→Save As</strong> to save the changes to another file.</td>
</tr>
<tr>
<td>Line</td>
<td>Displays the line position of the insertion point in the buffer.</td>
</tr>
<tr>
<td>Column</td>
<td>Displays the column position of the insertion point in the buffer.</td>
</tr>
<tr>
<td>Bytes</td>
<td>Shows the size of the buffer in bytes.</td>
</tr>
<tr>
<td>Modified</td>
<td>Indicates whether the buffer has unsaved changes.</td>
</tr>
</tbody>
</table>
Compile menu

The Compile menu options allow you to run and compile ABL procedures. Table 4–12 describes the menu that appears when you choose this option.

Table 4–12: Compile menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>Compiles and runs the procedure in the current buffer</td>
</tr>
<tr>
<td>Check Syntax</td>
<td>Checks the syntax of the procedure in the current buffer</td>
</tr>
<tr>
<td>Debug</td>
<td>Invokes the Debugger for the procedure in the current buffer</td>
</tr>
<tr>
<td>Compiler Messages</td>
<td>Displays the compiler messages for the most recent compilation</td>
</tr>
</tbody>
</table>
Compile>Run

Choose this option to compile and run a procedure in the current buffer. Figure 4–14 shows an example of the screen output of a compiled procedure.

<table>
<thead>
<tr>
<th>Name</th>
<th>Balance</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub Par Golf</td>
<td>19,675.00</td>
<td>37,000</td>
</tr>
<tr>
<td>SC Ren Je Rot</td>
<td>10,065.00</td>
<td>13,000</td>
</tr>
<tr>
<td>Olympique marseille</td>
<td>33,284.00</td>
<td>45,600</td>
</tr>
<tr>
<td>Batter Up Baseball</td>
<td>1,949.00</td>
<td>7,500</td>
</tr>
<tr>
<td>Purjehdustavike Oy</td>
<td>1,025.00</td>
<td>7,000</td>
</tr>
<tr>
<td>Jortsin Resport</td>
<td>18,955.00</td>
<td>23,000</td>
</tr>
<tr>
<td>Chip’s Poker</td>
<td>14,192.00</td>
<td>20,100</td>
</tr>
<tr>
<td>Butternut Squash Inc</td>
<td>25,964.00</td>
<td>26,800</td>
</tr>
<tr>
<td>Hangon Potkula Kosta</td>
<td>7,402.00</td>
<td>11,800</td>
</tr>
<tr>
<td>Offside Hockey</td>
<td>7,851.00</td>
<td>15,000</td>
</tr>
<tr>
<td>Spokes Cycles</td>
<td>850.00</td>
<td>10,000</td>
</tr>
<tr>
<td>Stard Snork Snorkel</td>
<td>1,657.00</td>
<td>15,000</td>
</tr>
<tr>
<td>Super Golf Center</td>
<td>11,805.00</td>
<td>12,000</td>
</tr>
<tr>
<td>Flying Fat Aerobics</td>
<td>7,616.00</td>
<td>10,700</td>
</tr>
<tr>
<td>Holvin juoksukeskus</td>
<td>519.00</td>
<td>5,500</td>
</tr>
<tr>
<td>Hou Hoog die Bal</td>
<td>2,949.00</td>
<td>10,500</td>
</tr>
<tr>
<td>Squash Meetarihalli</td>
<td>27,999.00</td>
<td>32,300</td>
</tr>
<tr>
<td>Hearts Darts</td>
<td>11,400.00</td>
<td>24,000</td>
</tr>
<tr>
<td>Luoppioisten Biljardi</td>
<td>9,517.00</td>
<td>11,800</td>
</tr>
<tr>
<td>Hannakeskinneys KY</td>
<td>2,459.00</td>
<td>6,000</td>
</tr>
<tr>
<td>Lagt Kort Ligger</td>
<td>40,439.00</td>
<td>51,700</td>
</tr>
<tr>
<td>First Down Football</td>
<td>76,504.00</td>
<td>82,200</td>
</tr>
<tr>
<td>Stay Afloat Swimming</td>
<td>12,581.00</td>
<td>13,200</td>
</tr>
<tr>
<td>UFO Frisbee</td>
<td>7,052.00</td>
<td>8,000</td>
</tr>
<tr>
<td>Sukellusvarustus</td>
<td>9,771.00</td>
<td>16,200</td>
</tr>
<tr>
<td>U-Jump Parachuting</td>
<td>14,225.00</td>
<td>21,500</td>
</tr>
<tr>
<td>Birdy’s Badminton</td>
<td>28,442.00</td>
<td>52,300</td>
</tr>
<tr>
<td>Auffi Bergausrustung</td>
<td>17,673.00</td>
<td>24,800</td>
</tr>
<tr>
<td>Hard Knocks Skating</td>
<td>38,424.00</td>
<td>46,400</td>
</tr>
<tr>
<td>Kempeleen Intersport</td>
<td>11,518.00</td>
<td>18,900</td>
</tr>
<tr>
<td>StickyWicket Cricket</td>
<td>34,863.00</td>
<td>54,800</td>
</tr>
<tr>
<td>Off The Wall</td>
<td>3,742.00</td>
<td>6,200</td>
</tr>
<tr>
<td>Second Skin Scuba</td>
<td>28,115.00</td>
<td>38,700</td>
</tr>
<tr>
<td>Fallen Arch Running</td>
<td>68,716.00</td>
<td>73,500</td>
</tr>
<tr>
<td>Spike’s Volleyball</td>
<td>18,287.00</td>
<td>20,400</td>
</tr>
</tbody>
</table>

Figure 4–14: Compiled procedure example output to screen

When the procedure is complete, press SPACEBAR to return to the Procedure Editor. You return to the buffer, and the cursor is in the same position as when you executed Compile→Run.

If you run a procedure that contains only the QUIT statement, the procedure stops running and the Procedure Editor automatically exits. If you have buffers open with changes in them, the Procedure Editor prompts you to save the changes before exiting.

Compile>Check Syntax

Choose this option to compile the procedure in the current buffer and check for ABL syntax errors.

The Compiler Messages dialog box explains any problems in the syntax. OpenEdge displays an alert box when the syntax is correct.
Compile>Debug

Choose this option to access the Debugger and debug the procedure code in the current buffer. For more information about debugging a procedure, see *OpenEdge Development: Debugging and Troubleshooting*.

**Note:** If the procedure in the current buffer does not compile, the Procedure Editor does not open the Debugger when you choose this option.

Compile>Compiler Messages

Choose this option to display the compiler messages for the most recent compile. Figure 4–15 shows an example of the Compiler Messages dialog box.

![Compiler Messages dialog box](Figure 4–15: Compiler Messages dialog box)

Tools menu

The Tools menu options allow you to access other tools. See Chapter 1, “Application Development Environment” for more information about this menu.

Help menu

The Help menu options allow you to access online help information about OpenEdge error messages and key mappings. See Chapter 1, “Application Development Environment” for more information about this menu.
Application Compiler

The OpenEdge Application Compiler utility lets you compile a set of source procedures (.p files) either for the duration of an OpenEdge session (or until OpenEdge runs out of directory entries) or for permanent storage. For a session, OpenEdge creates temporary r-code files. For permanent storage, OpenEdge creates r-code files that inherit the name of the source file and a .r extension by default. Once you compile a procedure, it does not get recompiled when you run it, so the procedure runs quickly. For more information on r-code, see *OpenEdge Getting Started: ABL Essentials*.

This chapter contains the following sections:

- Startup parameters for Compiler behavior
- Using the Application Compiler
- Compiling source files
Startup parameters for Compiler behavior

OpenEdge provides two startup parameters that may affect the way the source code is processed when you compile or check syntax: Compile Warning List (\(-cwl\)) and Keyword Forget List (\(-k\)). These options are briefly described here. For more information, see *OpenEdge Deployment: Startup Command and Parameter Reference*.

**Compile Warning List (-cwl)**

The Compile Warning List startup parameter specifies a set of ABL statements that trigger a warning message at compile time if they are found in the source code. The list can include any ABL statements you choose; it can be an effective way to discourage the use of deprecated statements. The occurrence of warnings does not prevent valid code from compiling.

Each warning appears in the following format:

```
The flagged_statement statement from the compile warning list was found in file file_name at line line_number.
```

If you see one or more such messages when you compile your code, click **OK** to continue, and if necessary, edit the code as appropriate.

**Keyword Forget List (-k)**

The Keyword Forget List startup parameter specifies a set of ABL keywords that the Compiler should not treat as keywords when processing the source code. This option allows you to avoid compilation errors that might otherwise occur if an upgraded version of OpenEdge introduces new keywords that are found in existing code as user-defined element names (such as tables, fields, frames, variables, streams, and so on).

If you encounter errors of this sort, you can use a Keyword Forget List to disable selected keywords, and thereby allow the code to compile, until you can remove those words from your application.
Using the Application Compiler

Choose **Tools → Application Compiler** from the menu bar of any other OpenEdge tool to start the Application Compiler.

When you start the Application Compiler, OpenEdge displays the window shown in **Figure 5–1**.

![Application Compiler window](image)

**Figure 5–1:** Application Compiler window

You use the Application Compiler window to select options and define settings for your compilation.

**Using the files/directories selection list**

The files/directories selection list displays the files or directories you want to compile. The default list contains only a period (.) indicating the OpenEdge working directory. You can modify the file and directory specifications using the action buttons.
Using the toggle boxes

The toggle boxes let you specify criteria for compiling your procedures. The following list describes the toggle boxes and how selecting them affects your compilation:

- **Save New .r Files** — Table 5–1 describes how choosing this toggle box affects the compilation process.

Table 5–1: Save New .r toggle box and the compilation process

<table>
<thead>
<tr>
<th>Action</th>
<th>R-code file</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not select</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save New .r Files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does not exist</td>
<td></td>
<td>Creates a temporary r-code file that lasts only for the duration of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>current OpenEdge session or until OpenEdge runs out of directory entries</td>
</tr>
<tr>
<td>Exists</td>
<td></td>
<td>Displays the error message Compile aborted. SAVE not specified and .r file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exists. The message informs you there already is a valid r-code file,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and the source file is not compiled unless you specify Save New .r Files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or Remove Old .r Files.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Select Save New .r Files</th>
<th>Does not exist</th>
<th>Compiles the source file and saves the new r-code file. If errors occur during compilation, no r-code file is created.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exists</td>
<td>Compiles the source file; the new r-code file replaces the existing r-code file. If errors occur during compilation, no r-code file is created, and the existing r-code file remains unchanged.</td>
</tr>
</tbody>
</table>

If you specify a directory in the **Save Into** field of the **Compiler Options** dialog box, the Application Compiler saves the new r-code files into that directory. If you do not specify a directory in the **Save Into** field, the Application Compiler saves the new r-code files into the directory that contains the source code files.

- **Look in Subdirectories** — Searches for and compiles the specified source files in the subdirectories of the selected directory.
Using the Application Compiler

- **Remove Old .r Files** — Deletes any existing r-code file that corresponds to the specified source file in the same directory. You do not need to select this option if you are going to save the new r-code files because newly compiled r-code files automatically replace existing ones.

  You must select this option when you want to compile source files for only the OpenEdge session (that is, without saving the r-code files). Thereafter, when you run the compiled procedures, the Application Compiler ensures that you are running the latest versions. If you do not first remove the existing r-code files, the Application Compiler does not recompile the source files. Instead, the error message Compile aborted. SAVE not specified and .r file exists appears.

- **Only Compile if No .r File** — Compiles the specified source files only when no corresponding r-code files are found in the selected directory.

**Note:** You cannot specify both **Remove Old .r Files** and **Only Compile if No .r File** at the same time. When you choose one option, the Compiler automatically deactivates the other.

**Using the action buttons**

The action buttons let you modify the file and directory selection list and compile your files. Table 5–2 describes the action buttons in the Application Compiler window.

<table>
<thead>
<tr>
<th>Button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propath</td>
<td>Displays all the directories that are located on your PROPATH. Highlight the directories you want to add to the file/directory selection list and choose OK.</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a file or directory outside of the PROPATH to the file/directory selection list. Specify the file or directory name you want to add.</td>
</tr>
<tr>
<td>Modify</td>
<td>Modifies the currently selected file or directory specification. Specify the change you want to make to the selected file or directory specification.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the currently selected file or directory from the file/directory selection list. Verify that you want to delete the selected file or directory from the selection list.</td>
</tr>
<tr>
<td>Start Compile</td>
<td>Compiles all the files specified in the file/directory selection list.</td>
</tr>
</tbody>
</table>
When you choose the **Add** or **Modify** action buttons, the **File Specification** dialog box shown in Figure 5–2 appears.

![File Specification dialog box](image)

**Figure 5–2: File Specification dialog box**

Table 5–3 describes the user-interface elements of the **File Specification** dialog box.

<table>
<thead>
<tr>
<th>User-interface element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>File or directory name</td>
<td>Specifies the name of the file or directory you want to add or modify.</td>
</tr>
<tr>
<td>Types</td>
<td>Specifies the filter the Application Compiler uses to perform a match on other strings such as file or directory names. A filter frequently uses a wildcard character to match strings, for instance, *a.* returns all filenames beginning with the letter a and containing a period. The default types are *.p and *.w, which return all file or directory names with the .p and .w extensions. The AVM evaluates filters with the MATCHES function. To specify more than one type, separate the types with a space; do not separate them with a comma.</td>
</tr>
<tr>
<td>Files</td>
<td>Displays the <strong>Files</strong> dialog box, which lists the files in the currently selected directory.</td>
</tr>
</tbody>
</table>

### Using the menu bar

The following sections describe the pull-down menus and the options available on the menu bar in the Application Compiler. **Table 5–4** describes the menu bar options.

<table>
<thead>
<tr>
<th>Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Exits the Application Compiler</td>
</tr>
<tr>
<td>Compile</td>
<td>Compiles ABL source files</td>
</tr>
<tr>
<td>Tools</td>
<td>Accesses other OpenEdge tools</td>
</tr>
<tr>
<td>Options</td>
<td>Accesses Compiler options and lets you save your Compiler settings</td>
</tr>
<tr>
<td>Help</td>
<td>Accesses the OpenEdge Help system</td>
</tr>
</tbody>
</table>
File menu

The File menu lets you exit the Application Compiler. Table 5–5 describes the menu that appears when you choose this option.

Table 5–5: File menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit</td>
<td>Ends your current Application Compiler session</td>
</tr>
</tbody>
</table>

Compile menu

The Compile menu lets you compile ABL source files. Table 5–6 shows the Compile menu.

Table 5–6: Compile menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Compile</td>
<td>Compiles source files</td>
</tr>
</tbody>
</table>

When you select the Compile → Start Compile option, the Compiler Results dialog box shown in Figure 5–3 appears.

Figure 5–3: Compiler Results dialog box

The dialog box displays messages about the status of the compilation. If you do not choose Options → Show Status, the Compiler only indicates when the compilation completes. If you choose Options → Show Status, however, the Compiler displays all messages generated during the compilation. For example, the messages state whether a file compiled successfully or whether it failed, and if so, on what line. Other messages include notices of incompatible CRC, and so on.

Tools menu

The Tools menu lets you access other tools. See Chapter 1, “Application Development Environment” for more information.
Options menu

The Options menu allows you to specify file information and display all the compilation status messages. Table 5–7 lists the items on the Options menu.

Table 5–7: Options menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler...</td>
<td>Supplies more information about the source files to be compiled</td>
</tr>
<tr>
<td>Show Status</td>
<td>Shows all messages in the Compiler Results window when you compile</td>
</tr>
</tbody>
</table>

Choose Options→Compiler to supply more information about the file you want to compile. When you choose this option, the Compiler Options dialog box appears as shown in Figure 5–4.

![Compiler Options dialog box](image)

Figure 5–4: Compiler Options dialog box

Table 5–8 describes the fields in the Compiler Options dialog box.

Table 5–8: Compiler Options dialog box fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default File Spec</td>
<td>Specifies the types of files you want to add to the file/directory specification list. The default file specification is *.p, *.w.</td>
</tr>
<tr>
<td>Message Log File</td>
<td>Specifies the file to which the Application Compiler sends compiler messages and status. The default output filename is compile.log.</td>
</tr>
<tr>
<td>Save into</td>
<td>Specifies the directory where you want to save the r-code files. By default, the Application Compiler saves the r-code files into the directory that contains the source file.</td>
</tr>
<tr>
<td>Languages</td>
<td>Identifies the language segment to be created in the r-code. The Application Compiler compiles the source file as well as strings translated into different language segments into the r-code file. The default language is the language in your source code. Enter the names of the languages separated by a space. ABL stores translated character strings for each specified language in separate text segments within the r-code file. For the Application Compiler to include translations, you must be connected to a translation database.</td>
</tr>
</tbody>
</table>

(1 of 4)
Table 5–8: Compiler Options dialog box fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>V6Frame</td>
<td>Selects a V6Frame option (Box, Underline, or Reverse Video) or accepts the default (No). V6Frame causes OpenEdge to compile and run Progress Version 6 applications on the latest version of OpenEdge. If you activate both the V6FRAME and STREAM-IO options, the STREAM-IO option overrides the V6FRAME option. For more information, see the COMPILE statement in <em>OpenEdge Development: ABL Reference</em>.</td>
</tr>
<tr>
<td>Stream-IO</td>
<td>Specifies that all output from the compiled procedure be formatted for output to a file or printer. All font specifications are ignored and all frames are treated as if they included the USE-TEXT option.</td>
</tr>
<tr>
<td>Listing File</td>
<td>Specifies the name of the listing file. The Application Compiler produces a listing of the compilation that includes the following information:</td>
</tr>
<tr>
<td></td>
<td>• The number of the block where each statement belongs</td>
</tr>
<tr>
<td></td>
<td>• The name of the source file you are compiling</td>
</tr>
<tr>
<td></td>
<td>• The date and time at the start of the compilation</td>
</tr>
<tr>
<td></td>
<td>• The line number of each line in the source file</td>
</tr>
<tr>
<td></td>
<td>• The complete text of all include files and the name of any subprocedures</td>
</tr>
<tr>
<td></td>
<td>• The buffer and frame scopes to procedures, internal procedures, and trigger blocks</td>
</tr>
<tr>
<td>Append (Listing File)</td>
<td>Specifies to append the current listing of the compilation to an existing listing file. If you do not select this option, the Application Compiler creates a new listing file that replaces any file of the same name.</td>
</tr>
<tr>
<td>Page Width</td>
<td>Specifies the page width for the listing file. The default page width is 80 characters per line. Enter a number between 80 and 255.</td>
</tr>
<tr>
<td>Page Length</td>
<td>Specifies the page length for the listing file. The default page length is 60 lines per page. Enter a number between 10 and 127.</td>
</tr>
<tr>
<td>XML Format</td>
<td>Specifies that cross-reference information between source files and database objects is to be written in XML format, rather than unformatted text, for improved readability. If you select this option, the label of the next field changes from Xref File to Xref File/Dir, and the Append (Xref File) option is disabled.</td>
</tr>
</tbody>
</table>

**Note:** The XML Schema used with these XML output files is stored in the following location: $DLC/properties/schemas/xrefdxxx.xsd. The xxxx portion of the file name indicates the version number of the file.
### Table 5–8: Compiler Options dialog box fields (3 of 4)

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Xref File (Xref File/Dir) | Specifies the file where the Application Compiler writes cross-reference information between source files and database objects. If you do not select the XML Format option, for each object reference, the xref file contains one unformatted and blank-separated line containing the following:  
- Procedure name  
- Source filename  
- Line number  
- Reference type  
- Object identifier  
If you select the XML Format option, when compiling a single procedure or class, you can provide a file name for the XML output file. This output file is overwritten each time the compiler needs to compile a linked class or procedure.  
When compiling several procedures and classes, supply a directory name. The Application Compiler uses this directory and a standard naming convention to capture the cross-reference information from multiple procedures and classes in separate files. Contrast this with the APPEND mode used by the Xref file in standard text format.  
When a directory is supplied, the compiler takes the root name of the procedure or class being compiled and creates a cross-reference file with this name and a .xref.xml file extension (sourcefilename.xref.xml). It stores it in the directory path specified, creating any subdirectories needed. If the main directory specified does not exist, an error will occur.  
See the COMPILe statement in OpenEdge Development: ABL Reference for more information on the cross-reference file. |
| Append (Xref File) | Specifies to append the xref listing of the source file to an existing xref file. If you do not select this option, the Application Compiler creates a new xref file that replaces any file of the same name. This option is disabled if you select the XML Format option. |
| Debug File | Specifies the file where the Application Compiler writes a listing to the debug file. The debug file consists of a line-numbered listing of the procedure with the text of all preprocessor include files, names, and parameters inserted. |
| Encryption Key | Specifies the key the Application Compiler uses during compilation to decrypt the source file and any encrypted include files. Encrypted files provide security against users accessing and modifying source files. OpenEdge lets you use either the default key to encrypt source procedures or your own encryption key. If you use your own encryption key, enter it here. If you use an encryption key, the Application Compiler does not produce a listing file as a security measure.  
For more information on encryption keys and procedures, see the COMPILe statement in OpenEdge Development: ABL Reference and the XCODE utility in OpenEdge Deployment: Managing ABL Applications. |
Help menu

The **Help** menu options allow you to access online help information about the Application Compiler. See Chapter 1, “Application Development Environment” for more information.

---

**Table 5–8: Compiler Options dialog box fields (4 of 4)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize R-code Size</td>
<td>Allows you to enter Yes to generate r-code with the minimum executable size or accept the default (No). An r-code file comprises multiple segments of varying length. During execution, r-code segments are swapped in and out of memory as required. However, not all segments are required in a deployment environment. Only the Application Debugger uses the debugger segment, and only the Open Client Proxy Generator (ProxyGen) uses the signature descriptor data. Typically, the end user never invokes the Application Debugger or ProxyGen as part of running their application. Thus, if you are deploying this r-code to an appropriate environment, you can use this option to deploy it with the smallest possible footprint.</td>
</tr>
<tr>
<td>Generate MD-5</td>
<td>Allows you to enter Yes to generate, and store in the r-code, a 128-bit MD5 value that WebClient™ can use to determine if the r-code file has changed since the previous version of the application. Progress Software Corporation recommends setting this option to Yes when compiling WebClient application procedures. Only WebClient uses the resulting MD5 value.</td>
</tr>
<tr>
<td>Defaults</td>
<td>Changes the current Compiler options to the Application Compiler default settings.</td>
</tr>
</tbody>
</table>
Compiling source files

This section explains how to compile ABL source files.

To compile source files with the Application Compiler:

1. Specify the source files you want to compile using the action buttons. You can select files and directories from your $PROPATH using the Propath button or from any other directories using the Add button.

2. Specify the compilation criteria using the toggle boxes.

3. Choose Options→ Compiler.

4. Specify any additional configuration information in the fields.

5. Choose Compile→ Start Compile or press F1 to compile the source files.

   The Compiler Results dialog box appears and displays messages indicating either the success or the failure of the compilation.

6. Choose OK from the Compiler Results dialog box to return to the Application Compiler window.
Index

A

About Tool option
  Help menu 1–7

Accelerator keys 1–4

Action buttons
  Application Compiler 5–5

ADE. See Application Development Environment

_adeevnt.p 3–2

Application Compiler
  action buttons 5–5
  Compile menu 5–7
  compiler options 5–8 to 5–11
  File menu 5–7
  files/directories selection list 5–3
  menu bar 5–6
  Options menu 5–8
  starting 1–3, 5–3
  toggle boxes 5–4
  Tools menu 1–6

Application Development Environment
  Application 1–2

Asterisk (*)
  marker 4–12

Attributes
  buffer 4–13

B

Buffer menu
  Procedure Editor 4–12

Buffers
  current buffer name 2–3
  filename 4–13
  information 4–13
  inserting fields 4–8
  inserting files 4–8
  listing 2–5, 4–12
  save as option 4–5
  settings 4–13
  size limits 2–4
  switching 2–5

Bytes
  buffer setting 4–13

C

Closing
  files 4–4

Column
  buffer setting 4–13

Command keys
  character Procedure Editor 2–3

Compile menu
  Application Compiler 5–7
  compiler messages option 4–16
  Procedure Editor 4–14

Compile Warning List
  startup parameter 5–2
Compiler messages 4–16

Compiling criteria
look in subdirectories 5–4
only compile if no .r files 5–5
remove old .r files 5–5
save new .r files 5–4
source files 5–12
startup parameters affecting 5–2

Cursor 2–9

D

Data Dictionary
Tools menu 1–6

Debug file
compiler option 5–10

Default file spec
Compiler option 5–8

Dialog boxes
buffer list 4–12
compiler options 5–8
compiler results 5–7
files 4–4
open files 4–3

E

Edit menu
Procedure Editor 4–7

Editor. See Procedure Editor

Encryption keys
compiler option 5–10

Exiting
files 4–4

F

File menu
Application Compiler 5–7
Procedure Editor 4–3
save as option 4–5

Filename
buffer setting 4–13

Files
closing 4–4
compiling 5–12
debug 5–10
exiting 4–6
listing 5–9
message log 5–8
opening 4–3
xref 5–10

Files/directories selection list 5–3

Find option
Search menu 4–9

G

Generate MD-5
compiler option 5–11

H

Help
about Tool option 1–7
menu 1–7

I

Information option
Buffer menu 4–13

Insert fields option
Edit menu 4–8

Insert file option
Edit menu 4–8

Insertion point 2–9
Procedure Editor 2–3

Integration hooks 3–1

K

Key
encryption 5–10

Keyboard option
Help menu 1–7

Keyword Forget List
startup parameter 5–2

Keywords, ABL
disabling 5–2
triggering usage alerts 5–2
L
Languages
  compiler option 5–8
Limits
  buffer size 2–4
Line
  buffer setting 4–13
Listing
  buffers 4–12
Listing file append
   compiler option 5–9
Listing files
  compiler option 5–9
Look in subdirectories
  toggle box 5–4

M
Markers
  asterisk 4–12
  current buffer 4–12
Menu bar
  Application Compiler 5–6
  Procedure Editor 4–2
Menus
  Application Compiler
   Compile 5–7
   File 5–7
   Option 5–8
   Help 1–7
  Procedure Editor
   Buffer 4–12
   Compile 4–14
   Edit 4–7
   File 4–3
   Search 4–9
   Tools 1–6, 5–7
Message log file
  compiler option 5–8
Messages
  compiler 4–16
Messages option
  Help menu 1–7
Minimize R-code Size
  compiler option 5–11
Modifying
  buffer settings 4–13
Multiple users
  startup 1–2, 2–2

O
Only compile if no .r files
  toggle box 5–5
Opening files 4–3
Options menu
  Application Compiler 5–8
OS shell option
  Tools menu 1–6

P
Page length
  compiler option 5–9
Page width
  compiler option 5–9
Procedure Editor 4–1
  Buffer menu 4–12
  buffer size limits 2–4
  Compile menu 4–14
  cursor 2–9
  Edit menu 4–7
  File menu 4–3
  Help menu 1–7
  insertion point 2–3
  menu bar 4–2
  Search menu 4–9
  starting 1–2, 2–2
  Tools menu 1–6
  window layout 2–2

R
R-code file 5–1
Recent messages option
  Help menu 1–7
Remove old .r files
  toggle box 5–5
Replace option
S

Save as option
   File menu 4–5
Save into
   compiler option 5–8
Save new .r files
   toggle box 5–4
Search menu
   Procedure Editor 4–9
Settings
   buffer 4–13
Single-user
   startup 1–2, 2–2
Source code management tools 3–1
Source files 5–1
   compiling 5–12
Starting
   Application Compiler 1–3, 5–3
   OpenEdge 2–2
   Procedure Editor 1–2, 2–2
Starting OpenEdge 1–2

Startup parameters
   for Application Compiler 5–2
Stream-IO option
   Application Compiler 5–9

T

Text
   find option 4–9
   replace option 4–10
Text cursor. See cursor

Toggle boxes
   look in subdirectories 5–4
   only compiler if no .r files 5–5
   remove old .r files 5–5
   save new .r files 5–4

Tools menu 1–6

X

Xref file
   append
      compiler option 5–10
      compiler option 5–10