OpenEdge® Web Paper: Batch-mode Event Support
Notices

© 2014 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

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

Business Empowerment, Business Making Progress, Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Empowerment Center, Fathom, Making Software Work Together, OpenEdge, Powered by Progress, Progress, Progress Control Tower, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress RPM, Progress Software Business Making Progress, Progress Software Developers Network, Rollbase, RulesCloud, RulesWorld, SequeLink, SpeedScript, Stylus Studio, Technical Empowerment, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, AppsAlive, AppServer, BusinessEdge, Progress Easy!, DataDirect Spy, DataDirect SupportLink, Easy!, Future Proof, High Performance Integration, OpenAccess, Pacific, ProDataSet, Progress Arcade, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress Pacific, Progress Profiles, Progress Results, Progress RFID, Progress Responsive Process Management, Progress Software, ProVision, PSE Pro, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the Release Notes applicable to the particular Progress product release for any third-party acknowledgments required to be provided in the documentation associated with the Progress product.

The Release Notes can be found in the OpenEdge installation directory and online at: https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx.

For the latest documentation updates see OpenEdge Product Documentation on Progress Communities: (https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx).

August 2014

Last updated with new content: Release 10.1A
Table of Contents

Preface..................................................................................................................................................7
  Purpose................................................................................................................................................7
  Using this manual.................................................................................................................................7
  References to ABL compiler and run-time features..........................................................................8
  References to ABL data types..........................................................................................................8
  Typographical conventions...........................................................................................................8

Chapter 1: Batch-mode Event Support.........................................................................................11
Preface

For details, see the following topics:

• Purpose
• Using this manual
• Typographical conventions

Purpose

This web paper provides an overview of the events supported by OpenEdge® running in batch-mode.

Using this manual

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

For the latest documentation updates see the OpenEdge Product Documentation Overview page on Progress Communities:

References to ABL compiler and run-time features

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

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

References to ABL data types

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

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

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

Typographical conventions

This manual uses the following typographical and syntax 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><strong>Italic</strong></td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
</tbody>
</table>
## Convention

<table>
<thead>
<tr>
<th><strong>Concertion</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>A plus sign between key names indicates a <strong>simultaneous</strong> key sequence: you press and hold down the first key while pressing the second key. For example, <strong>CTRL+X</strong>.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>A space between key names indicates a <strong>sequential</strong> key sequence: you press and release the first key, then press another key. For example, <strong>ESCAPE H</strong>.</td>
</tr>
</tbody>
</table>

### Syntax:

- **Fixed width**: A fixed-width font is used in syntax, code examples, system output, and file names.
- **Fixed-width italics**: Fixed-width italics indicate variables in syntax.
- **Fixed-width bold**: Fixed-width bold italic indicates variables in syntax with special emphasis.
- **UPPERCASE fixed width**: ABL keywords in syntax and code examples are almost always shown in upper case. Although shown in uppercase, you can type ABL keywords in either uppercase or lowercase in a procedure or class.
- **Period (.) or colon (:)**: All statements except **DO**, **FOR**, **FUNCTION**, **PROCEDURE**, and **REPEAT** end with a period. **DO**, **FOR**, **FUNCTION**, **PROCEDURE**, and **REPEAT** statements can end with either a period or a colon.
- **[ ]**: Large brackets indicate the items within them are optional.
- **[]**: Small brackets are part of ABL.
- **{ }**: Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.
- **{}**: Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.
- **|**: A vertical bar indicates a choice.
- **...**: Ellipses indicate repetition: you can choose one or more of the preceding items.
Batch-mode Event Support

OpenEdge running in batch mode supports the following events:

**Note:** AppServer™ servers are in essence batch mode ABL clients, so batch mode event support also applies to AppServers.

- PROCEDURE-COMPLETE
- READ-RESPONSE
- CONNECT
- Developer events (U1 to U10) and CLOSE

There are ten miscellaneous events (U1 to U10 developer events) that have no standard meaning or action. They are reserved for an ABL developer to use for associating a trigger with some event that only happens programmatically using the APPLY ABL statement, and which has nothing to do with a particular user action. These developer events have no built-in significance. There is also the CLOSE event which has these same characteristics. The CLOSE event only occurs when you specify it with the APPLY statement in your application.

You can use developer events or CLOSE in your application when running an OpenEdge client in batch mode. This means an application can execute the WAIT-FOR ABL statement with a developer or CLOSE event in batch mode. Similarly, you can specify ON "U1" OF or ON "CLOSE" OF an object and then programmatically APPLY "U1" to the object to execute the code. There is no change in how OpenEdge process these events. That is, OpenEdge clients running in batch mode process these events just like non-batch clients do.
An example of when using developer events in batch mode could be useful is when you have an application that works with ABL sockets, asynchronous AppServer requests, or asynchronous state-free AppServer requests. For instance, an application might spawn several asynchronous AppServer requests, and use one or more developer events to determine when all requests completed, as opposed to using the `PROCEDURE-COMPLETE` event in a loop until all requests complete. If working with a state-free AppServer, the application will get the response to the asynchronous requests out of order and you can use these developer events to better handle these situations.