OpenEdge® Getting Started: New and Revised Features
# Table of Contents

## Preface

- Purpose ............................................................................................................................................ 7
- Audience ........................................................................................................................................... 8
- Organization ................................................................................................................................... 8
- Using ABL documentation ..................................................................................................................... 8
  - References to ABL compiler and run-time features ................................................................. 8
  - References to ABL data types ..................................................................................................... 9
- Typographical conventions .................................................................................................................. 9
- Examples of syntax descriptions ........................................................................................................ 10
  - Long syntax descriptions split across lines ............................................................................. 12
  - Complex syntax descriptions with both required and optional elements .................................. 12
- Example procedures ......................................................................................................................... 12
- OpenEdge messages .......................................................................................................................... 13
  - Obtaining more information about OpenEdge messages ......................................................... 14

## Chapter 1: What's New in 11.7

- Installation and platform support ...................................................................................................... 16
- Startup parameters ............................................................................................................................. 16
- OpenEdge Change Data Capture ........................................................................................................ 18
- OpenEdge ABL enhancements ........................................................................................................... 19
  - Asynchronous request improvements ....................................................................................... 19
  - Change Data Capture API support ............................................................................................ 20
  - Class-based object improvements ............................................................................................... 20
  - Color Coding in 64-bit Windows Development Tools ............................................................... 21
  - Compiler options phrase ............................................................................................................ 21
  - Debugger breakpoint on error improvements ............................................................................. 21
  - Handle-based object validation .................................................................................................... 22
  - Handling STOP conditions using stop objects .......................................................................... 22
  - JSON improvements .................................................................................................................... 23
  - Temp-table improvements ........................................................................................................... 24
  - WebClient improvements .............................................................................................................. 25
- OpenEdge Authentication Gateway ....................................................................................................... 25
- OpenEdge Management and OpenEdge Explorer .............................................................................. 26
- OpenEdge RDBMS .............................................................................................................................. 28
- OpenEdge Replication ......................................................................................................................... 31
- OpenEdge Server Technology ............................................................................................................. 32
- OpenEdge SQL .................................................................................................................................... 34
- OpenEdge DataServers ......................................................................................................................... 35
Chapter 2: 11.7 Feature Comparisons.................................................................41
   Spring Security Configuration Changes in PAS for OpenEdge ..............................41

Chapter 3: Release 11.7 Documentation.............................................................45
   Changes to the documentation set........................................................................45
   Accessing OpenEdge documentation....................................................................46
   Release 11.7 documentation set...........................................................................46
Preface

For details, see the following topics:

• Purpose
• Audience
• Organization
• Using ABL documentation
• Typographical conventions
• Examples of syntax descriptions
• Example procedures
• OpenEdge messages

Purpose

OpenEdge Getting Started: New and Revised Features briefly describes both new features and changes to existing features introduced in OpenEdge® Release 11.7.0, and directs you to where you can find more detailed information about these changes in the documentation set. The Release 11.7 documentation set references include:

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

This guide is primarily intended for OpenEdge application developers and system administrators who are upgrading their license to Release 11.7 from Release 11.6.

Organization

What's New in 11.7 on page 15

Presents brief explanations of new and enhanced features introduced in Release 11.7, and references places within the documentation set where more detailed feature information can be found.

11.7 Feature Comparisons on page 41

Presents brief explanations of features whose behavior has changed since Release 11.6.

Release 11.7 Documentation on page 45

Provides information about the OpenEdge Release 11.7 documentation set.

Using ABL documentation

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 documentation 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><em>Italic</em></td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, <strong>GET</strong> and <strong>CTRL</strong>.</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.
### Convention

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPPERCASE fixed width</td>
<td>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.</td>
</tr>
<tr>
<td>Period (.) or colon (;)</td>
<td>All statements except \texttt{DO, FOR, FUNCTION, PROCEDURE, and REPEAT} end with a period. \texttt{DO, FOR, FUNCTION, PROCEDURE, and REPEAT} statements can end with either a period or a colon.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Large brackets indicate the items within them are optional.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Small brackets are part of ABL.</td>
</tr>
<tr>
<td>{ }</td>
<td>Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.</td>
</tr>
<tr>
<td>{}</td>
<td>Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>. . .</td>
<td>Ellipses indicate repetition: you can choose one or more of the preceding items.</td>
</tr>
</tbody>
</table>

### Examples of syntax descriptions

In this example, \texttt{ACCUM} is a keyword, and \texttt{aggregate} and \texttt{expression} are variables:

#### Syntax

```
ACCUM aggregate expression
```

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

```
FOR EACH Customer NO-LOCK:
  DISPLAY Customer.Name.
END.
```

In this example, \texttt{STREAM stream}, \texttt{UNLESS-HIDDEN}, and \texttt{NO-ERROR} are optional:

#### Syntax

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```
In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

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

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

Long syntax descriptions split across lines

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

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

Syntax

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

Complex syntax descriptions with both required and optional elements

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

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

Syntax

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

Example procedures

OpenEdge documentation may provide example code that illustrates syntax and concepts. You can access many of the example files, and details for installing them, from the following locations:

• A self-extracting Documentation and Samples file available on the OpenEdge download page of the Progress Software Download Center
Once installed, you can locate the example files in the following paths under the OpenEdge Documentation and Samples installation directory:

<table>
<thead>
<tr>
<th>This directory . . .</th>
<th>Contains examples for the following documents . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>src\prodoc\dotnetobjects</td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
<tr>
<td>src\prodoc\dynamics</td>
<td>The Progress Dynamics documentation</td>
</tr>
<tr>
<td>src\prodoc\getstartoop</td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td>src\prodoc\handbook</td>
<td>OpenEdge Getting Started: ABL Essentials</td>
</tr>
<tr>
<td>src\prodoc\interfaces</td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>src\prodoc\json</td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
<tr>
<td>src\prodoc\langref</td>
<td>OpenEdge Development: ABL Reference</td>
</tr>
<tr>
<td>src\prodoc\prodatasets</td>
<td>OpenEdge Development: ProDataSets</td>
</tr>
<tr>
<td>src\prodoc\tranman</td>
<td>OpenEdge Development: Translation Manager</td>
</tr>
<tr>
<td>src\prodoc\visualdesigner</td>
<td>OpenEdge Getting Started: Introducing Progress Developer Studio for OpenEdge Visual Designer</td>
</tr>
<tr>
<td>src\prodoc\xml</td>
<td>OpenEdge Development: Working with XML</td>
</tr>
<tr>
<td>src\samples\open4gl\java</td>
<td>OpenEdge Development: Java Open Client</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.

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

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

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

**Obtaining more information about OpenEdge messages**

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

• Choose Help > Recent Messages to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.

• Choose Help > Messages and then type the message number to display a description of a specific OpenEdge message.

• In the Procedure Editor, press the HELP key or F1.

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

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

1. Start the Procedure Editor:

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

2. Press F3 to access the menu bar, then choose Help > Messages.

3. Type the message number and press ENTER. Details about that message number appear.

4. Press F4 to close the message, press F3 to access the Procedure Editor menu, and choose File > Exit.
What's New in 11.7

This chapter summarizes new and enhanced features in OpenEdge® Release 11.7. For more detailed information about a feature, see the OpenEdge documentation that is referenced.

For details, see the following topics:

- Installation and platform support
- Startup parameters
- OpenEdge Change Data Capture
- OpenEdge ABL enhancements
- OpenEdge Authentication Gateway
- OpenEdge Management and OpenEdge Explorer
- OpenEdge RDBMS
- OpenEdge Replication
- OpenEdge Server Technology
- OpenEdge SQL
- OpenEdge DataServers
- Progress Application Server for OpenEdge
- Progress Developer Studio for OpenEdge
- OpenEdge Business Process Management
• OpenEdge Application Development Environment
• Technology Preview for STOP condition handling using stop objects

Installation and platform support

OpenEdge release 11.7 installation and platform support includes the following:

• **Configutil utility** — The configutil utility helps you to export the source installation configurations and import them to a different OpenEdge installation on the same or different machine.

• **Authentication Gateway** — The **OpenEdge Authentication Gateway Details** dialog box lets you set the configuration details for the OpenEdge Authentication Gateway (Security Token Service) to be used with Progress Application Server for OpenEdge.

• **OpenEdge upgrades to JAVA 8** — OpenEdge now uses Java 1.8.0_101.

• **Infragistics upgrade** — OpenEdge release 11.7.0 uses Infragistics NetAdvantage for .NET v2016 Vol 2. The Infragistics controls are upgraded to version 16.2.20162.2040.

• **OpenEdge upgrades to .NET 4.6** — OpenEdge now uses 4.6.

• **Downloadable License Configuration File** — When obtaining the software product installation kits from the Progress Electronic Software Delivery (ESD) system, customers now have the option of using a self-service portal to generate and download license configuration (.cfg) files based on either a subset of products in a single License Addendum ("green sheet") or a combination of products across multiple License Addendums. A license configuration file can then be added to a previously-installed product image without the need for a re-install. This feature is supported for OpenEdge Release 10.2B and later.

For more information, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Getting Started: Installation and Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Install Help</td>
</tr>
</tbody>
</table>

Startup parameters

OpenEdge Release 11.7 includes new startup parameters, described in the following table.

**Table 1: New Startup Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>For more Information see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>-baseuserindex</td>
<td>Specify the first index for which statistics are collected in the _UserIndexStat VST.</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td>-baseusertable</td>
<td>Specify the first table for which statistics are collected in the _UserTableStat VST.</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
</tbody>
</table>
For more information, including default values, see:

**Manuals:**
- OpenEdge Deployment: Startup Command and Parameter Reference
- OpenEdge Data Management: Database Administration
- OpenEdge Getting Started: OpenEdge Authentication Gateway Guide

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>For more Information see...</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-catchStop n</code></td>
<td>Use the Catch STOP (<code>-catchStop</code>) startup parameter to specify whether the AVM activates stop object behavior when a STOP condition is raised.</td>
<td>See Handling STOP conditions using stop objects on page 22</td>
</tr>
<tr>
<td><code>-cdcsiz</code></td>
<td>Specify the size of the CDC cache.</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td><code>-noclr</code></td>
<td>(PAS for OpenEdge only) Use the No Common Language Runtime (<code>-noclr</code>) startup parameter to disable the Common Language Runtime (CLR) component, which disables the execution of .NET programs.</td>
<td>OpenEdge Deployment: Startup Command and Parameter Reference</td>
</tr>
<tr>
<td><code>-numcheckpointstats</code></td>
<td>Specify the number of rows maintained in the _Checkpoint VST.</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td><code>-recspacesearchdepth</code></td>
<td>Specify how far into the record free chain the database engine searches for a block with enough free space to store the data associated with a record create or update before a new cluster of data is allocated.</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td><code>-secsize</code></td>
<td>Specify the size of the security cache.</td>
<td>OpenEdge Getting Started: OpenEdge Authentication Gateway Guide</td>
</tr>
</tbody>
</table>

OpenEdge Release 11.7 includes updated startup parameters, described in the following table.

**Table 2: Updated Startup Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>For more Information see...</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-D n</code></td>
<td>The default Directory Size (<code>-D</code>) setting for compiled procedure directory entries has been changed from 100 to 1000 entries.</td>
<td>OpenEdge Deployment: Startup Command and Parameter Reference</td>
</tr>
<tr>
<td><code>-s n</code></td>
<td>The default Stack Size (<code>-s</code>) setting has been changed from 40 to 128.</td>
<td>OpenEdge Deployment: Startup Command and Parameter Reference</td>
</tr>
</tbody>
</table>
OpenEdge Change Data Capture

OpenEdge Release 11.7 includes the new feature, OpenEdge Change Data Capture (CDC).

Change Data Capture (CDC) is an industry term that describes the process of duplicating subsets of OLTP data in an external data source with a relatively up to date version of relational data. The OpenEdge implementation of CDC provides a flexible and scalable capture process to facilitate the data extraction, transformation, and eventually the loading of the data to an external data source. The data provided by the capture process exists on the source database and is maintained in relational form.

OpenEdge CDC is flexible because:

- Captured data is maintained in the same database
- Captured data is maintained in relational form
- Captured data is accessible with SQL and ABL

OpenEdge CDC is scalable because you can define CDC policies such that:

- The amount of data captured is variable by table. You can capture no data, some data, or the whole record.
- You can index the data for easier retrieval.
- The amount of data captured is controlled through policies defined at the table and field level.

Benefits of OpenEdge Change Data Capture include:

- It identifies and tracks all changes made within the OpenEdge RDBMS
- It guarantees accurate tracking of all data changes regardless of where they occur
- It increases efficiencies and availability of changes for ETL to sync identified changes with other data sources, data repositories or data warehouses
- Has a single point of configuration regardless of ABL or SQL
- It can be activated with zero changes to the application, just configure and run
- It can be managed completely online—no downtime required

CDC is a licensed product layered on top of a Workgroup or Enterprise RDBMS license.

Configuration of CDC is controlled through OpenEdge Management or OpenEdge Explorer. See OpenEdge Management and OpenEdge Explorer on page 26 for more details.

Support for CDC is provided in both SQL and ABL. See OpenEdge SQL on page 34 and OpenEdge ABL enhancements on page 19 for more details.
OpenEdge ABL enhancements

OpenEdge Release 11.7 adds support for the ABL features described in the following sections.

- Asynchronous request improvements on page 19
- Change Data Capture API support on page 20
- Class-based object improvements on page 20
- Color Coding in 64-bit Windows Development Tools on page 21
- Compiler options phrase on page 21
- Debugger breakpoint on error improvements on page 21
- Handle-based object validation on page 22
- Handling STOP conditions using stop objects on page 22
- JSON improvements on page 23
- Temp-table improvements on page 24
- WebClient improvements on page 25

Asynchronous request improvements

OpenEdge Release 11.7 supports the following asynchronous request improvements:

- **ProDataSet parameter support** — In previous releases, you cannot invoke a procedure asynchronously if the procedure has a ProDataSet parameter. In this release, this limitation is removed. As with other parameter data types, for INPUT-OUTPUT and OUTPUT ProDataSet parameters passed to an asynchronous request, the asynchronous event procedure receives these parameters as INPUT parameters.

- **Asynchronous request object handle** — This object handle has two new attributes:
  - **ERROR-OBJECT** — This attribute returns an object reference to an instance of a class that implements the Progress.Lang/Error interface and is thrown from an asynchronous procedure or Web service operation. This instance describes either an ERROR condition thrown as an error object or a STOP condition.
for certain errors thrown as a `Progress.Lang.StopError` object. Using this attribute to access a `Progress.Lang.StopError` object is enabled only if you enable the new stop object handling feature in the AVM.

- **STOP-OBJECT** — When enabled, this attribute returns an object reference to an instance of the `Progress.Lang.Stop` class, or one of its built-in subclasses, that is thrown from an asynchronous procedure or Web service operation. This instance describes a STOP condition for supported application features. This attribute can only have a valid object reference if you enable the new stop object handling feature in the AVM.

For more information on using these attributes when stop object handling is enabled, see [Handling STOP conditions using stop objects](#) on page 22.

For more information on these improvements, see:

| Manual: | OpenEdge Development: ABL Reference |

---

### Change Data Capture API support

OpenEdge Release 11.7 provides new OpenEdge API and utility classes in support of Change Data Capture (CDC):

- **OpenEdge.DataAdmin** package class and enumeration API to access and manage the CDC schema information — This API provides access to the CDC schema information through the `OpenEdge.DataAdmin.DataAdminService` class and allows ABL code to perform typical operations, such as create, read, edit, and delete for configuring CDC table and field policies.

- **OpenEdge.DataAdmin.Util** package utility classes and enumerations — Utility classes are added to assist developers writing programs to retrieve data from the CDC Change Tracking tables.

For more information, see:

| Manual: | OpenEdge Development: Programming Interfaces |

---

### Class-based object improvements

OpenEdge Release 11.7 supports the following class-based object improvements:

- **Run-time performance** — This release supports performance improvements for run-time class management. For example, instead of loading all built-in classes at startup, they are loaded on demand, as each class needs to be accessed.

- **Object serialization improvements** — A new NON-SERIALIZABLE option is now available in class member definitions for properties, ProDataSets, temp-tables, and variables, which suppresses serialization of these members for classes that are otherwise marked as SERIALIZABLE. Also, in addition to its existing availability as an option and attribute setting for defining ProDataSets and temp-tables as class data members, `SERIALIZE-NAME` is now available as an option for specifying an alternate name for serializing a defined class property or a variable defined as a class data member that is not also marked as NON-SERIALIZABLE.

For more information, see:

| Manuals: | OpenEdge Development: ABL Reference |
| | OpenEdge Development: Object Oriented Programming |
Color Coding in 64-bit Windows Development Tools

The Procedure Editor and AppBuilder Section Editor in 64-bit Windows products now support syntax color coding. Most of the features of the color-coding editor in 32-bit products are supported in this release, including UTF-8 support, syntax expansion, aliases, auto-casing of keywords, code reformatting (indent/unindent, comment/uncomment), and configuration dialog boxes.

For more information, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Development: AppBuilder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>AppBuilder Help</td>
</tr>
<tr>
<td></td>
<td>Procedure Editor Help</td>
</tr>
</tbody>
</table>

Compiler options phrase

OpenEdge Release 11.7 supports a new COMPILE statement options phrase that enforces one or more rules during compilation. This phrase is designed to provide an extensible mechanism for specifying compiler requirements that apply to different compilation environments.

In this release, rules you can specify include disallowing abbreviated field names (require-full-names), disallowing abbreviated keywords (require-full-keywords), and disabling implicit resolution of buffer references (require-field-qualifiers).

The following options allow you to specify these requirements:

- OPTIONS — The rules are specified by a character expression that evaluates to a comma-separated list of options.
- OPTIONS-FILE — The rules are specified in the same way, but supplied in a designated file.

For more information, see:

| Manual:                      | OpenEdge Development: ABL Reference |

Debugger breakpoint on error improvements

In OpenEdge Release 11.7, both the standalone OpenEdge Debugger and the Debugger provided with the Progress Developer Studio for OpenEdge ensure that all breakpoints triggered for an error break on the line where the error occurred, rather than at a line where execution branches after the error, as in previous releases. In addition, all error breakpoints now recognize and break on errors thrown using the ABL UNDO, THROW statement.

For more information on debugging with error breakpoints, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Development: Debugging and Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Progress Developer Studio for OpenEdge Online Help</td>
</tr>
</tbody>
</table>
Handle-based object validation

In OpenEdge Release 11.7, when you pass a handle (instead of a class-based object reference) to the \texttt{VALID-OBJECT} function, this function works in the same way as the \texttt{VALID-HANDLE} function.

For more information, see:

| Manual: | OpenEdge Development: ABL Reference |

Handling STOP conditions using stop objects

OpenEdge Release 11.7 adds the ability in ABL to handle STOP conditions by catching and throwing certain class-based objects (stop objects). This feature is thus analogous to the support in previous releases to handle ERROR conditions by catching and throwing error objects. As with handling error objects using CATCH blocks and handling ERROR conditions using the \texttt{ON ERROR} phrase, you can combine the handling of stop objects using CATCH blocks with the handling of STOP conditions using the \texttt{ON STOP} phrase.

In addition, using CATCH blocks to handle STOP conditions, you can now identify different types of STOP conditions depending on the type of stop object that is caught, and you can trap any error messages associated with the STOP condition, which was not possible in previous releases.

Note: In OpenEdge Release 11.7, support for handling STOP conditions as stop objects is a available as a "Technology Preview" only. For more information, see Technology Preview for STOP condition handling using stop objects on page 39.

In order to use this stop object feature in an OpenEdge session, you must explicitly enable it by setting the Catch STOP (-catchStop) startup parameter to 1 (see Startup parameters on page 16). By default, the feature is disabled and any use of stop objects in the session are ignored by the AVM.

You must recompile any existing source code that contains a CATCH block for Progress.Lang.Error, Progress.Lang.SysError, or Progress.Lang.ProError, and of course, you must compile any code that references any of the new stop objects. The compilation will succeed regardless of the value of the -catchStop startup parameter. However, without -catchStop 1 on at run time, the new stop handling behavior will not take effect.

Following are descriptions of the ABL elements that have been added and updated to support stop object handling.

These are the new ABL built-in classes that implement stop objects. When enabled, the AVM throws an appropriate stop object for every STOP condition that it raises:

- \texttt{Progress.Lang.Stop} — The AVM throws an instance of this class when the STOP statement executes. This class inherits from Progress.Lang.Object and also provides a single public CallStack property that works similar to the CallStack property provided for error objects. This is also the immediate super class for the following built-in objects that the AVM throws in response to the associated STOP conditions:

  - \texttt{Progress.Lang.LockConflict} — The AVM throws an instance of this class when a time-out occurs while waiting for a record lock or when the user presses Cancel in the dialog box displayed during the record lock waiting period. This class also provides public Device, TableName, and User properties to identify the device, database table, and user that holds the record lock.

  - \texttt{Progress.Lang.StopAfter} — The AVM throws an instance of this class when a time-out occurs in response to the STOP-AFTER setting on a DO, FOR, or REPEAT block.
• **Progress.Lang.UserInterrupt** — The AVM throws an instance of this class when the user interrupts the AVM by pressing the **STOP** key, which is usually mapped to **CTRL−C** on Unix/Linux or **CTRL−Break** on Windows.

• **Progress.Lang.StopError** — The AVM throws an instance of this class when the AVM encounters certain unexpected error conditions. Although thrown for a **STOP** condition, objects of this class inherit from **Progress.Lang.SysError** and therefore implement the **Progress.Lang.Error** interface, allowing them to return error messages and a call stack associated with the error that raised the **STOP** condition.

The following ABL syntax has been updated to work with stop objects when enabled:

• **CATCH statement** — Updated to catch any specified stop object, as well as any error object, including any stop object thrown from an OpenEdge application server to an OpenEdge client.

  **Note:** For any **Progress.Lang.StopError** object thrown on an OpenEdge application server, the associated error messages are written to the server log file regardless if the object is caught and handled on the server or the client. This differs from **STOP** condition handling without stop objects enabled, as in previous releases, where any associated error messages for **STOP** conditions raised on the server are always written to the server log file and never reach the client.

• **DO, FOR, and REPEAT statements** — Updated to throw a **Progress.Lang.StopAfter** object when a **STOP−AFTER** time-out setting is satisfied and raises the **STOP** condition.

• **ON STOP phrase** — Updated to work with any **CATCH** block for a stop object that is coded in the associated block on which the **ON STOP** phrase is specified. Note that a **CATCH** block for a stop object takes precedence over any **ON STOP** phrase that applies to the same associated block.

• **STOP statement** — Updated to throw a **Progress.Lang.Stop** object when it raises the **STOP** condition.

• **UNDO, THROW statement** — Updated to allow a stop object, as well as an error object, to be thrown.

The following new attributes of the asynchronous request object handle, when enabled, return any associated stop object thrown from the asynchronous request:

• **ERROR-OBJECT** — This attribute returns an object reference to any instance of the **Progress.Lang.StopError** class that is thrown from an asynchronous request.

• **STOP-OBJECT** — This attribute returns an object reference to any instance of the **Progress.Lang.Stop** class, or one of its built-in subclasses, that is thrown from an asynchronous request.

For more information on these attributes, see Asynchronous request improvements on page 19.

For more information on **STOP** condition handling using stop objects, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
</table>

**Note:** Although *OpenEdge Development: Error Handling* describes both **ERROR** and **STOP** condition handling in OpenEdge, this manual has not been updated to describe the stop object feature. For a more detailed overview of the feature in this release, see the description of the **CATCH** statement in *OpenEdge Development: ABL Reference*.

## JSON improvements

OpenEdge Release 11.7 supports the following improvements to JSON management in ABL:
Chapter 1: What's New in 11.7

- **Serialization and deserialization of Progress.Lang.Object fields** — Previous releases have supported the ability to serialize and deserialize ABL class-based objects to and from a JSON representation using methods of the built-in Progress.IO.JsonSerializer class. However, handle methods for reading and writing JSON representations of buffers, temp-tables, and ProDataSets were restricted from serializing and deserializing any Progress.Lang.Object fields for temp-tables.

  With this release, both the WRITE-JSON( ) and SERIALIZE-ROW( ) methods now serialize these fields for any buffers, temp-tables, and ProDataSets that they write to a JSON representation. Also, the READ-JSON( ) method deserializes these fields for any JSON representation that it reads into buffers, temp-tables, and ProDataSets.

- **Writing a JsonArray or JsonObject object target from an ABL data object** — Previous releases have supported the ability to read a Progress.Json.ObjectModel.JsonArray or Progress.Json.ObjectModel.JsonObject source into an appropriate buffer, temp-table, or ProDataSet object handle using its READ-JSON( ) method. However, writing to similar JsonArray and JsonObject targets using the WRITE-JSON( ) and SERIALIZE-ROW( ) methods was not supported.

  With this release, both the WRITE-JSON( ) and SERIALIZE-ROW( ) methods can now write to an appropriate Progress.Json.ObjectModel.JsonArray or Progress.Json.ObjectModel.JsonObject target when called on a corresponding buffer, temp-table, or ProDataSet object handle.

  This support includes changes to some of the options available when calling these methods for the new targets, for example, use of the omit-outer-object parameter when calling WRITE-JSON( ).

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
</tbody>
</table>

**Temp-table improvements**

OpenEdge Release 11.7 supports the following temp-table improvements:

- **Adding a before-table to a dynamic temp-table** — In previous releases, if you had a handle to a dynamic temp-table without a before-table created for it, you could not then explicitly create its before-table. OpenEdge implicitly creates a before-table in certain situations, such as setting the TRACKING-CHANGES attribute on the dynamic temp-table handle to TRUE. But this requires the temp-table to be part of a ProDataSet; otherwise, setting this attribute raises an error. Other mechanisms also create a before-table when the source for the temp-table (such as JSON) already has a before-table defined.

  With this release, you can now explicitly create a before-table for any dynamic temp-table that does not already have one by invoking the TEMP-TABLE-PREPARE( ) method on its object handle and passing it an optional LOGICAL parameter that when TRUE causes the AVM to create the before-table. This mechanism allows a dynamic temp-table to be prepared with a before-table when its source does not already have a before-table defined, such as a JSON source that was generated without one. This then allows the dynamic temp-table to be passed to an ABL class method, where the method parameter requires a temp-table argument that already has a before-table defined.

- **Calling EMPTY-TEMP-TABLE( ) on a temp-table handle** — In previous releases, you could only call the EMPTY-TEMP-TABLE( ) method on a buffer object handle.

  With this release, you can call the EMPTY-TEMP-TABLE( ) method on a temp-table object handle as well.
WebClient improvements

OpenEdge Release 11.7 provides a new option in the WebClient Application Assembler to continue application execution when the WebClient cannot update its application configuration.

In previous releases when first executing a WebClient application, the software checks the Web server that is hosting the application configuration (prowcapp) file to determine if an application update is available. If the Web server is not available, an error is generated and the WebClient terminates. There are situations where it is appropriate for the WebClient to continue execution, even if it is not able to contact the hosting Web server.

In this release, you can set an option in the WebClient Application Assembler to continue WebClient execution even if it cannot access an updated application configuration. This is the new Continue Offline check-box on the General tab. When selected, the WebClient application that you generate will continue execution if the WebClient cannot access an updated application configuration on the Web server.

For more information, see:

Manual: OpenEdge Deployment: WebClient Applications

OpenEdge Authentication Gateway

The OpenEdge Authentication Gateway is a key component of a centralized authentication and authorization service for database connections. It is an implementation of a Security Token Service (STS) that creates and seals the client-principal tokens that are passed to a web server for single sign-ons.

It is supported by other OpenEdge components, including:

• Utilities to configure OpenEdge databases to access OpenEdge Authentication Gateway services
• Tools to maintain and monitor OpenEdge database activity when using the Authentication Gateway
• ABL functions, handle attributes and methods, and class properties that support OpenEdge Authentication Gateway services

Note: In addition to ABL database connections, the OpenEdge Authentication Gateway also supports authentication for SQL database connections.

The Authentication Gateway is a secured Progress Application Server (PAS) for OpenEdge instance where the OESTS web application (oests.war) is deployed. The Authentication Gateway requires domain configuration and access to an authentication provider (for example: OS Local, LDAP, Active Directory, etc.) to be able to create and seal client-principal tokens.

For more information, see:

Manual: OpenEdge Getting Started: OpenEdge Authentication Gateway Guide
OpenEdge Management and OpenEdge Explorer

OpenEdge release 11.7 provides the following OpenEdge Management and OpenEdge Explorer features and improvements:

- **Change Data Capture management** — Change Data Capture (CDC) is a tracking mechanism that enables applications to determine the changes made to user tables in a database. Using CDC in OpenEdge, you can choose individual fields of a table for which you want to capture the changes and publish the data to an external data source. OpenEdge Management allows you to define and manage table and field policies for CDC.

  Using OpenEdge Management, you can:

  - Enable CDC for a database.
  - Create CDC policies for a new table and existing tables.
  - View a list of tables that are enabled for CDC, and CDC policies for each table.
  - Modify, activate, deactivate, and delete CDC policies.
  - Dump and load CDC policies.

- **Third-party Web application support** — OpenEdge Management supports custom third-party Web applications to accommodate management capabilities for Web applications beyond the OpenEdge environment.

  With support for third-party Web applications, you can:

  - Host your own Java Web applications in the OpenEdge Management Web server.
  - Integrate your own web pages.
  - Extend existing OpenEdge Management functionality.

- **New OpenEdge resource—Batch Program** — OpenEdge Management comes with a new OpenEdge resource, Batch Program, designed to run for longer periods of time and supports separate scripts to start, monitor, and stop the Program. Like any other resource, OpenEdge Management allows you to create, configure, and monitor Batch Program resources.

  Using OpenEdge Management, you can:

  - Create, start, stop, and delete Batch Program resources.
  - Configure properties for Batch program resources.
  - View and manage multiple processes that are starting, running, or stopped for a Batch Program.
  - View Batch Program resource log files using the log file viewer.
  - Setup monitoring plans and schedules for a Batch Program resource.

- **OpenEdge Authentication Gateway based authentication** — Along with property-file based authentication, OpenEdge Management now supports OpenEdge Authentication Gateway (OEAG) based authentication.

  **Note:** Creating, managing, and monitoring of Batch Program resources is not supported in OpenEdge Explorer.
OpenEdge Management provides an interface which allows you to select OEAG as the authentication mechanism to log into the OpenEdge Management console.

- **New rules for PAS for OpenEdge** — New rules are provided for PAS for OpenEdge monitoring plans. These rules check (poll) a PAS for OpenEdge instance according to the time interval specified and trigger alerts when the instance performs outside the defined threshold values. The threshold values range from 1 to 100 percent.

- **New diagnostic pages** — New Work Scheduler diagnostic pages are now available for reporting purposes. These pages include:
  
  - Resource Scheduler Details
  - Work Scheduler Queue
  - AdminServer Threads
  - Resource Activity History
  - Logging
  - System
  - Jobs

- **Landing page preferences** — You can now choose various options as the landing page you want to view after you log into your management console.

  In the management console, click **Options > User Preferences** and select one of the following as the landing page:

  - **Default landing page** — The default dashboard page.
  - **Selected view** — A specific view, either the default view created by the management console or a view created by you.
  - **Resource list** — The resource grid page with a list of all resources.
  - **Alerts list** — The alerts grid page with a list of all alerts.
  - **Selected resource home page** — The home page of a specific resource.
  - **Other** — A page from third-party Web application.

    This option is available only if a third-party Web application is hosted in the management console.

- **OpenEdge manager login failure** — When you create and start a PAS for OpenEdge instance with incorrect OpenEdge manager credentials, the OpenEdge manager login fails and the PAS instance home page displays a message, an alert, and the current status of the instance to notify you about the login failure.

- **REST and SOAP transport properties configuration** — You can now configure few properties for REST and SOAP transports of a local PAS for OpenEdge instance even when the OpenEdge manager web application, `oemanager`, is not installed and the instance is not running.

- **Upgrade requests for Web Server configuration** — When configuring OpenEdge Management Web server, an option, **Upgrade insecure HTTP requests to HTTPS**, is provided to automatically upgrade insecure connection requests to secure connection requests.

For more information, see:
OpenEdge RDBMS

OpenEdge Release 11.7 includes the following new features for the OpenEdge RDBMS:

- **Change Data Capture** — The RDBMS fully supports CDC. See OpenEdge Change Data Capture on page 18 for details.

- **Replication** — The RDMS supports enhancements to Replication. See OpenEdge Replication on page 31 for details.

- **Database Client Notification**

  Database Client Notification is a communication mechanism between clients and the database, informing clients of the need to refresh their schema cache. Notification enables a more timely completion of certain database administrative actions, such as activating an inactive index. Prior to OpenEdge Release 11.7, certain database utilities required clients to disconnect, even though the utility can execute while the database is online. The disconnect of the clients is required because these utilities update index schema. Database Client Notification allows clients to remain connected and refresh their schema in step with the utility execution. Notification enables the index activation processes to move to completion more rapidly.

  Utilities supported by database client notification are

  - PROUTIL IDXACTIVATE
  - PROUTIL LOAD (with buildindexes)
  - PROUTIL MTIDXBUILD
  - PROUTIL TPIDXBUILD

  Database client notification is controlled by a server startup parameter, -usernotifytime, that indicates how often a client polls for notification.

- **Monitoring enhancements**

  Updates have been made to many VSTs and PROMON to provide greater access to data that can help programmatically monitor a running database for tuning, diagnostics, and health checking purposes.

  The following table provides a list of fields added to existing VSTs. The VSTs are updated to provide greater access to data that can help programmatically monitor a running database for tuning, diagnostics, and health checking purposes.
<table>
<thead>
<tr>
<th>VST Table name</th>
<th>Field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ActBuffer</td>
<td>_Buffer-Active</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Area</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Control</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Free</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Index</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Master</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Object</td>
</tr>
<tr>
<td></td>
<td>_Buffer-ObjList</td>
</tr>
<tr>
<td></td>
<td>_Buffer-RM</td>
</tr>
<tr>
<td></td>
<td>_Buffer-Seq</td>
</tr>
<tr>
<td>_AreaStatus</td>
<td>_AreaStatus-AIActivated</td>
</tr>
<tr>
<td></td>
<td>_AreaStatus-AI-Seq</td>
</tr>
<tr>
<td></td>
<td>_AreaStatus-State</td>
</tr>
<tr>
<td></td>
<td>_AreaStatus-LastOp</td>
</tr>
<tr>
<td></td>
<td>_AreaStatus-Type</td>
</tr>
<tr>
<td>_Checkpoint</td>
<td>_Checkpoint-Cluster</td>
</tr>
<tr>
<td></td>
<td>_Checkpoint-Number</td>
</tr>
<tr>
<td>_Connect</td>
<td>_Connect-NumSeqBuffers</td>
</tr>
<tr>
<td></td>
<td>_Connect-NumTrans</td>
</tr>
<tr>
<td></td>
<td>_Connect-UsedSeqBuffers</td>
</tr>
<tr>
<td></td>
<td>_Connect-UserMisc</td>
</tr>
<tr>
<td>_DbStatus</td>
<td>_DbStatus-ZeroStatsDate</td>
</tr>
<tr>
<td>_Logging</td>
<td>_Logging-AISequence</td>
</tr>
<tr>
<td></td>
<td>_Logging-BIClusterHWM</td>
</tr>
<tr>
<td></td>
<td>_Logging-CurrBICluster</td>
</tr>
</tbody>
</table>
### Table of VSTs and Field Names

<table>
<thead>
<tr>
<th>VST Table name</th>
<th>Field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>_StatBase</td>
<td>_StatBase_BaseUserIndex</td>
</tr>
<tr>
<td></td>
<td>_StatBase_BaseUserTable</td>
</tr>
<tr>
<td>_UserIO</td>
<td>_UserIO-BiRecRead</td>
</tr>
<tr>
<td></td>
<td>_UserIO-BiRecWrite</td>
</tr>
<tr>
<td>_UserLock</td>
<td>_UserLock-HWM</td>
</tr>
<tr>
<td>_Trans</td>
<td>_Trans-BIRecReads</td>
</tr>
<tr>
<td></td>
<td>_Trans-BIRecWrite</td>
</tr>
<tr>
<td></td>
<td>_Trans-JTA-State</td>
</tr>
<tr>
<td></td>
<td>_Trans-JTA-Flags</td>
</tr>
<tr>
<td></td>
<td>_Trans-XID</td>
</tr>
</tbody>
</table>

Startup parameters to control the VSTs are also added:

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base user index (-baseuserindex)</td>
<td>Specify the first index for which statistics are collected in the _UserIndexStat VST.</td>
</tr>
<tr>
<td>Base user table (-baseusertable)</td>
<td>Specify the first table for which statistics are collected in the _UserTableStat VST.</td>
</tr>
<tr>
<td>Number of checkpoint statistics (-numcheckpointstats)</td>
<td>Specify the number of rows maintained in the _Checkpoint VST.</td>
</tr>
<tr>
<td>Record free chain search depth factor (-recspacesearchdepth)</td>
<td>Specify how far into the record free chain the database engine searches for a block with enough free space to store the data associated with a record create or update before a new cluster of data is allocated.</td>
</tr>
<tr>
<td>User index range (-userindexrangesize)</td>
<td>Specify the number of rows maintained in the _UserIndexStat VST.</td>
</tr>
<tr>
<td>User table range (-usertablerangesize)</td>
<td>Specify the number of rows maintained in the _UserTableStat VST.</td>
</tr>
</tbody>
</table>

- **DBTOOL** — DBTOOL is enhanced with a new option to fix a rare case of schema corruption in the _user table when converting a database with PROUTIL CONV1011 that has over 65535 rows in the _user table.
- **IDXCHECK** — PROUTIL IDXCHECK is enhanced to detect index corruption caused by indexes straddling the 32/64 bit boundary.
OpenEdge Replication

OpenEdge Release 11.7 includes the following new features for OpenEdge Replication:

• **Recovery Transition**
  Recovery Transition is enhanced through the implementation of Replication Sets. A Replication Set is a two-target Replication configuration where the two targets are aware of each other, and are configured to continue replication in the event of a server failure. The target replicas can transition together, with one becoming the source replica and the other becoming the target of the new source. Replication Sets minimize the risk of a single point of failure, and can also avoid the time consuming process to rebase.

• **Enhancements to VSTs and Replication Monitor**
  The replication VSTs and Replication Monitor are expanded to support Replication Sets, and the underlying communication support structure.

  Updated tables include:
  • _Repl-Server
  • _Repl-Agent
  • _Reply-AgentActivity
  • _Repl-AgentControlActivity
  A new VST table for replication is, _Repl-InterAgentActivity

• **DSRUTIL Status** — The DSRUTIL status command is updated to include a verbose option that returns a description of the status in addition to the return code.

• **Parameter values are enforced**
  Prior to 11.7, a warning was issued if the startup parameters for a target database were less than the source. Starting in 11.7, an error is returned, and the target database agent does not start. This case is enforced for the following parameters:
  • Lock Table Entries (-L)
  • Number of Users (-n)
  • Maximum JTA Transactions (-Maxxids)
  • The sum of Number of Users (-n) and Maximum Servers (-Mn)
OpenEdge Server Technology

OpenEdge Release 11.7 contains Server Technology updates to provide the following features:

• **Native Java Distribution (JSSE) Support** — Starting with OpenEdge 11.7, the default Transport Layer Security (TLS) network library for the following has changed from the RSA vendor to Native Java distribution (JSSE):
  - AppServer broker (client connections)
  - WebSpeed broker (client connections)
  - Java OpenClient (connection to application servers)
  - AIA adapter (connection to application servers)
  - WSA adapter (connection to application servers)
  - REST adapter (connection to application servers)

If the OpenEdge 11.7 JSSE library’s TLS restrictions do not allow network connections, you may temporarily revert to using the RSA library.

• **Wildcard Support** — You can add a wildcard "*" character in the subject name or common name fields in the public key certificate so that you can define a single common name for multiple domain names. For example, a certificate with "*.bedford.progress.com" common name can be used for user1.bedford.progress.com, user2.bedford.progress.com, or user3.bedford.progress.com.

• **Subject Alternate Name (SAN) Support** — You can now use Subject Alternative Name (SAN) to specify additional host names (values) to be protected by a single SSL certificate using a subjectAltName field. It allows more than one host to use the same copy of a single certificate. At the server-level, you can create multiple virtual hosts and add these hosts to the subjectAltName field of the certificate. You generate a certificate with SAN and the clients can connect to the server using subjectAltName. Whenever HTTPS request comes to any of the virtual host, the server uses the same certificate for SSL handshake.

• **Server Name Indication (SNI) Support** — You can now add Server Name Indication (SNI) by setting parameters in the client’s CONNECT() method. It allows the client to add the hostname that it attempts to connect to during the handshake as a part of the TLS negotiation. It enables the server to select the required domain name and present the certificate with the correct name. It also allows the server to present multiple certificates on the same IP address and TCP port number and thus allow multiple secure (HTTPS) websites (or any other Service over TLS) to be served off the same IP address without requiring all those sites to use the same certificate.

**Note:** OpenEdge supports SNI only for ABL Client and .NET Open Client.

• **HornetQ Support** — OpenEdge has certified the JMS-compliant vendor, HornetQ for generic Java Message Service (JMS) adapter.

• **sslj utility** — The sslj utility helps you to test and troubleshoot SSL or TLS communication and list the ciphers and protocols using the same Java JSSE library that is used by all Java based OpenEdge products.
such as classic Appserver, PAS for OE, classic WebSpeed, Java OpenClient, and OEM. The utility is located in $DLC/bin.

- **OpenSSL Library Upgrade**: OpenSSL Library is upgraded to 1.1.0c.

For more information on these topics, see:

<table>
<thead>
<tr>
<th>Manuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenEdge Application Server: Administration</td>
</tr>
<tr>
<td>OpenEdge Development: Messaging and ESB</td>
</tr>
<tr>
<td>OpenEdge Getting Started: Core Business Services - Security and Auditing</td>
</tr>
</tbody>
</table>
OpenEdge SQL

OpenEdge Release 11.7 contains OpenEdge SQL updates to provide the following features:

- **Authorized Data Truncation (ADT) - Output Option** – A new option, Output, was added to the settings for ADT. When the SQLTruncateTooLarge parameter is set to OUTPUT, SQL uses an alternate model of data truncation, in which data is allowed internally to exceed its defined size while data is operated upon at its actual, full size. When SQL outputs data to the application, SQL truncates any data that exceeds its size, to its defined size. This model provides consistency between SQL operations and the physical dimensions of database values. The application safely receives data in its logically defined size, after truncation. The default value for the SQLTruncateTooLarge parameter is OFF.

  **Note:** When the SQLTruncateTooLarge parameter is set to OUTPUT, all SQL operations, such as comparison and sorting, use actual data and only the output is truncated.

- **CDC** – OpenEdge SQL supports Change Data Capture (CDC), and implementations of CDC that can involve Extraction/Transformation/Load (ETL) applications that extract individual data changes for each table. Instead of dealing with the entire table, OpenEdge customers can selectively choose fields from an individual table to capture data and populate it to an external data source. This approach avoids replicating the entire database or an entire table, and provides scalability to the process. OpenEdge CDC is a scalable solution as it allows the CDC policy designer to determine the amount of data change detail needed.

  For ease of interpretation of meta-data fields like _Change-FieldMap that are of the type VARBINARY and represent whether the corresponding fields in the Change table have changed or not, OpenEdge SQL provides two scalar functions:
  
  - **CDC_get_changed_columns**
  - **CDC_is_column_changed**

- **JVM** – OpenEdge SQL uses a Java Virtual Machine (JVM) to execute Triggers, Stored Procedures, and User Defined Functions. Users can specify parameters which should be used by the JVM. JVM Parameters are the options that can be specified while starting Java using the command line. Users, with privileges to create or edit the `db_name.oesql.properties` files, can customize JVM parameters to control the behavior of the JVM. The JVM parameters used by SQL server are logged to the `db_name.lg` file.

- **UDF** – OpenEdge SQL provides support for User Defined Functions (UDF) that allow users to extend SQL functionality. A User Defined Function, written in JAVA, contains the logic, accepts the input parameters, and returns a scalar value as the result. User Defined Functions can help reduce network traffic. A complex set of logic that has the potential to filter out a large portion of result set rows, like a Regular Expression that cannot be represented as a single scalar expression, can be expressed as a User Defined Function. User Defined Functions allow users the advantage of having modularity in their code, and also provides the flexibility of reusing UDF.

- **Enhanced SQL Logging capabilities** – OpenEdge SQL now provides enhanced logging capabilities and control over logging to a granular level. SQL server has the provision to record logging information to the file `SQL_server_server-id_ddmmyyyy_hhmmss.log`. In this file name, the server-id corresponds to the server ID shown in the database log file (`db_name.lg`). While establishing a connection to the database, if logging is enabled and is set to a level between 1 and 4, users can find corresponding levels of detailed logging information in the log file. The level of detail in the log entry depends on the level of logging that is chosen. Level 1 logs the most basic and most important information. Increasing the logging level increases the level of detail that is logged to the log file.
OpenEdge DataServers

Progress OpenEdge release 11.7 provides the following OpenEdge DataServers features and improvements:

- **Statement level lock upgrade for JOIN query** — The OpenEdge MSS DataServer now supports statement level lock-upgrade for join query evaluated using server side join. OpenEdge ABL allows a user to specify a lock type in the `OPEN QUERY` statement of a `JOIN` query. An `EXCLUSIVE-LOCK` can be applied to individual buffers that are participating in a `JOIN` query. Opening a `JOIN` query when one or more participating buffers are opened with `EXCLUSIVE-LOCK` makes the DataServer evaluate the query to be joined by the client. Such join queries are not evaluated on the server and their performance might be affected. To use this capability, Progress suggests that users open a query with the lock-type `NO-LOCK`, and specify the desired locking at the statement level. This allows the DataServer to evaluate the query as server side join.

- **Join Robustness** — The OpenEdge DataServer for Oracle provides enhanced stability to server side `JOIN` capabilities. Dynamic join queries with ABL function references in the `WHERE` clause are optimized to be resolved as `JOIN-BY-SQLDB`.

- **Optimized ABL Cross Join (nested query) as JOIN-BY-SQLDB** — The OpenEdge DataServer for Oracle is enhanced to offload `CROSS JOIN` as server side join. By default, an ABL `CROSS JOIN` query is resolved on the ABL client. Server side join capability of an ABL `CROSS JOIN` is available based on the `QUERY-TUNING (JOIN-BY-SQLDB)` switch.

- **Oracle DataServer Compatibility with multi-tenant foreign databases** — The OpenEdge DataServer for Oracle is certified with the Oracle 12c multi-tenant configuration.

- **Microsoft SQL Server 2016 certification** — The OpenEdge MSS DataServer is now certified with Microsoft SQL Server 2016.

- **OpenEdge DataServer for ODBC is retired** — Starting with OpenEdge 11.7, the OpenEdge DataServer for ODBC is retired. The *Progress OpenEdge Data Management: DataServer for ODBC* guide has been removed from the product set. However, there is still conditional backwards compatibility in Progress networking mode.

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Data Management: SQL Development</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Data Management: SQL Reference</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Internationalizing Applications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>Data Management: DataServer for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Management: DataServer for Oracle</td>
</tr>
</tbody>
</table>
Progress Application Server for OpenEdge

New and revised features in PAS for OpenEdge include:

- **Apache Tomcat upgrade** — The core Apache Tomcat server is upgraded to Version 8.5.x in the 11.7 release of PAS for OpenEdge.

- **Name change** — In Progress OpenEdge 11.6 and earlier releases, PAS for OpenEdge was known as the Pacific Application Server for OpenEdge. In the 11.7 release, PAS is changed to Progress Application Server for OpenEdge.

- **Debugger support** — You can now use the Progress Developer Studio's Debugger to debug ABL business application code running in a PAS for OpenEdge server.


- **Multi-domain support** — Multi-domain support is extended to include implementations where the back-end system's user account storage does not specify domains or tenancy. The primary example of this is an LDAP authentication provider.

- **HTTP Client Single Sign-On (SSO) Support** — SSO support in the web server framework allows a user to log into the full business application once and transparently pass verifiable proof their login identity to each distributed web application (or service). While SSO can be implemented in the web application itself, the better choice is to implement the SSO through the web server framework. Using the web server framework for SSO has the benefit of executing the necessary security measures in a protected environment that the web application code cannot provide. Using web application framework SSO services also affords high availability and scalability support missing from web application implemented solutions.

- **Deployment architecture** — A new instance-name/ablapps directory where you can optionally deploy your web application's r-code for development and deployment.

- **New TCMAN create options** — The -m option allows you to specify a username and password to access Tomcat container-level security. The -Z option which allows you to specify a security model for web application during their development.

- **PASOESTART** — A new command-line utility encapsulates PAS for OpenEdge startup tasks into a single command.

- **genspringpwd** — A utility for generating encrypted passwords for the WEB-INF/users.properties user accounts file.

- **SSL Support Deprecated** — Due to the upgrade to Java 8 in OpenEdge 11.7, support for SSLv3 and other cipher suites (MD5 and RC5, for example) has been retired. Instead, support for TLS is updated, and replaces those retired technologies.

  **Note:** Nothing need be done to re-configure instances for TLS after completing an OpenEdge 11.7 install and the instance migration described in OpenEdge Getting Started: Migrating to OpenEdge 11.7. However, if you must revert to those retired technologies, you cannot use OpenEdge 11.7 or later releases.

- **WSASP Not Supported** — The wsasp.dll, that allowed ASP calls through IIS, is no longer supported. It will not work in OpenEdge 11.7.

For more information on these topics, see:
Progress Developer Studio for OpenEdge

OpenEdge Release 11.7 contains Progress Developer Studio for OpenEdge updates to provide the following features:

- **New OpenEdge Project wizard**—The interface for the OpenEdge Project wizard has changed. The Create an OpenEdge Project page now displays Server, Client, and General categories for project type selection. Do one of the following:
  - To create a project for the back-end development of the application, select the Server category from the left section of the page. Select the server type that you want to deploy the services to and the transport type that you want to use for deploying the application.
  - To create a project for the front-end development of the application, select the Client category from the left section of the page. Select the Desktop Application or the Web Application option button to create a project for a desktop application or a web application respectively and select one of the project types.
  - To create a basic, ABLUnit, Dynamics, or custom OpenEdge project, select the General category from the left section of the page.

  **Note:** If you do not select a category (Client, Server, or General) and click Finish, a basic OpenEdge project without any specialization is created.

- **Debugger support**—You can now use the Debug Configurations wizard to debug ABLUnit and Progress Application Server for OpenEdge.

- **Log Viewer**—The Log Viewer helps you to view and monitor the log files of a PAS for OpenEdge server instance from the Servers view. There are multiple log files for a server, you can choose to go to the directory where the log files are saved and manually search for errors, or view from Developer Studio using the Log Viewer.

- **ANT tasks for Progress Application Server for OpenEdge**—You can write custom ANT tasks to package ABL Web App and Web UI projects and to generate a Data Object Service Catalog file.

- **Migration of Spring Security files**—If you imported an ABL Web App project from OpenEdge 11.6 to the current release in the Developer Studio, you can also migrate the Spring Security files.

- **Infragistics controls upgrade**—The Infragistics controls are upgraded to version 16.2.20162.2040.

- **Telerik UI for WinForms controls upgrade**—The Telerik UI for WinForms controls that installs by default is upgraded to R3 2016 SP1(2016.3.1024). If you need a different version, you can download it from the Telerik web site.

- **Eclipse version update**—The Eclipse version is updated to 4.5.2.

For more information on these topics, see:

| Online help: | Progress Developer Studio for OpenEdge |
OpenEdge Business Process Management

Progress OpenEdge release 11.7 provides the following OpenEdge BPM features and improvements:

• **Web service adapter configuration using server-config.xml** — When defining the security settings for Web service adapter, you can now configure the Authentication details and Endpoint URL using the `server-config.xml` file.

• **New samples** — The following sample apps are now available for OpenEdge BPM:
  
  • **OEBPM JSDO Service (Classic & PASOE)** — A Data Object service implementation of OpenEdge BPM for both Classic AppServer and Progress Application Server for OpenEdge.
  
  • **OEBPM KUIB Sample** — A sample app to create new BPM forms.

  **Note:** In Progress Developer Studio, click Help > Samples to access these sample apps in the Samples page.

• **Removal of the Form Editor** — The Form Editor has been removed from BPM projects.

  In earlier releases, the Form Editor enabled you to design a customized interface with Form as the presentation format for Activity worksteps (and for Start worksteps in Business Processes). This format in the Portal presentation type has been removed for both Business Processes and Web applications.

• **Kendo UI Builder integration** — The following improvements have been made to integrate Kendo UI® Builder by Progress® (Kendo UI Builder) with OpenEdge BPM:

  • **Create BPM forms** — You can use in-built templates of Kendo UI Builder to create new BPM forms in OpenEdge 11.7

  • **Semantic Annotations for JSDO catalog** — You can now use semantic annotations to validate the dataslots supported by the JSDO catalog that is used with the Kendo UI Builder components.

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>Progress OpenEdge Business Process Server: Managed Adapters Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Getting Started: Developing BPM Applications with Developer Studio</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Online help:</th>
<th>Progress OpenEdge BPM Studio User Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Progress OpenEdge Business Process Modeler User Guide</td>
</tr>
<tr>
<td></td>
<td>Progress OpenEdge Business Process Server Documentation</td>
</tr>
<tr>
<td></td>
<td>Progress OpenEdge Business Process Portal User's Guide</td>
</tr>
</tbody>
</table>

OpenEdge Application Development Environment

OpenEdge release 11.7 provides the following OpenEdge ADE features and improvements:
Change Data Capture support — OpenEdge Database tools, Data Dictionary and Data Administration, provide support for managing Change Data Capture (CDC) and its reporting, and other database administration utilities.

Using these Database tools, you can:

- Check if a database is enabled for CDC.
- View CDC policy information in reports.
- Dump and load CDC policies.
- Create a DELETE trigger for a CDC change table.

In Audit Policy Maintenance tool, when selecting the tables for auditing, you will not find the CDC change tables as they cannot be enabled for auditing.

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: Basic Database Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td>Online help:</td>
<td>Data Administration Help</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary Help</td>
</tr>
<tr>
<td></td>
<td>Audit Policy Maintenance Help</td>
</tr>
</tbody>
</table>

Technology Preview for STOP condition handling using stop objects

STOP condition handling using stop objects is a feature that is supplied as a “Technical Preview” in OpenEdge Release 11.7, as defined in the product End User License Agreement (EULA):

**Technical Preview Features** — Product(s) may include features that are supplied as a “Technical Preview”, and if included, such features are documented in the Product documentation such as the Product Releases Notes. Progress is not obligated to provide any maintenance, technical or other support for the Technical Preview features, but may, from time to time, provide technical support and/or Updates to the Technical Preview features at Progress’ sole discretion. All such Updates shall be subject to the terms and conditions of this Agreement, including, without limitation, the terms and conditions set forth in this section. NOTWITHSTANDING ANYTHING TO THE CONTRARY IN SECTION 3 OF THE END USER LICENSE AGREEMENT, TECHNICAL PREVIEW FEATURES, ANY UPDATES THERETO AND ANY RELATED TECHNICAL SUPPORT SERVICES ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. PROGRESS MAKES NO WARRANTIES WITH RESPECT TO THE TECHNICAL PREVIEW FEATURES, ANY UPDATES THERETO, OR ANY RELATED TECHNICAL SUPPORT SERVICES, WHETHER EXPRESS, IMPLIED, OR ARISING BY CUSTOM OR TRADE USAGE, AND SPECIFICALLY MAKES NO WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT.

For more information on this feature and its documentation, see Handling STOP conditions using stop objects on page 22.
11.7 Feature Comparisons

This chapter provides comparisons of how features in Release 11.7 differ from previous releases.

For details, see the following topics:

- Spring Security Configuration Changes in PAS for OpenEdge

Spring Security Configuration Changes in PAS for OpenEdge

The changes in Spring Security configuration in PAS for OpenEdge in the 11.7 release are designed to make the framework more extensible, easier to configure, easier to administer, and easier to maintain from release to release.

These design goals are achieved by the implementation of a new security properties file (oeablSecurity.properties), and a new URL access control file oeablSecurity.csv. Basically, these new files replace configuration of the bean constructor and property settings that you would need to search for in multiple Spring Security XML files. These new "external" properties files are designed to be release independent, to be open to external management, and to provide access to all of the security process functionality in the Spring Security framework.
The security properties file

For some configurations, the new `oeablSecurity.properties` file employs a naming convention `(bean-name.property=value)` that simplifies the identification of a bean and its properties. For example:

```
OEClientPrincipalFilter.sealAnonymous=false
```

In other cases, a `property=value` pair replaces a reference to a specific XML configuration file. For example, in 11.6 you would reference the configuration template, `/WEB-INF/oeablSecurity-basic-local.xml`, in the `web.xml` file if you wanted to specify basic HTTP authentication using a local users file.

In 11.7, you don't edit the `web.xml` file, nor do you reference or edit a specific XML file. Instead, basic authentication with a local source for user information is specified by this set of properties in the `oeablSecurity.properties` file:

```
http.all.authmanager=local
client.login.model=basic
```

**Note:** Some Spring Security XML configuration files have been changed for the 11.7 release. However, the functionality that existed in 11.6 also can be found in 11.7. (The Spring Security XML files have been moved to the `WEB-INF/spring` directory on the server.) The goal was to refactor the XML files to eliminate the need to edit them directly.

In addition, you should be aware that the `oeablSecurity.properties` file can exist in three-tiered hierarchy so that you can:

1. Configure properties that are used for all ABL applications deployed on the instance by setting properties in `instance-name/conf/oeablSecurity.properties`. The defaults established in this file can be overridden by properties set in an ABL application (#2) or a web application (#3).

2. Configure properties that are used for ABL applications by setting properties in `instance-name/ablapps/abl-app-name/oeablSecurity.properties`. Property settings in this file override the server instance defaults (#1).

   **Note:** This level is optional. It is useful when you have more than one ABL application deployed on an instance and when those ABL applications require different security configurations.

3. Configure properties for a specific web application in an ABL application by setting properties in `instance-name/webapps/web-app-name/WEB-INF/oeablSecurity.properties`. Property settings in this file override the server instance defaults (#1) and the ABL application defaults (#2).

**URL access control file**

The `instance-name/webapps/web-app-name/WEB-INF/oeablSecurity.csv` file is added in 11.7 to implement URL access controls for web applications. In releases prior to 11.7, URL access controls were defined as `intercept-url` elements in `oeablSecurity-XXXX-XXXX.xml` configuration files.

Access control lists, since they are ordered sets of three values, do not lend themselves well to the format of a properties file with its name/value pairs. Therefore, URL access controls were not included in the `oeablSecurity.properties` file. CSV files are more suitable for access control lists, and they are easily maintainable by many external administrative tools.
The three fields in oeablSecurity.csv file correspond directly with the three attributes of an 11.6 intercept-url element, which are:

- **pattern** — the URL pattern which can include wildcards and regular expressions
- **method** — the HTTP access method (optional in 11.6, but required in 11.7)
- **access** — role[s] that are allowed access to the resource

For example, an intercept-url element would be specified like the following snippet from an 11.6 oeablSecurity-basic-local.xml configuration file:

```xml
<intercept-url
    pattern="/web/sales/**"
    method="GET"
    access="hasAnyRole('ROLE_PSCAdmin','ROLE_PSCUser')"/>
```

The snippet above grants access to any user who has either ROLE_PSCAdmin or ROLE_PSCUser privileges to data from a resource whose URL begins with /web/sales/.

An equivalent entry in the new 11.7 oeablSecurity.csv file looks like the following:

```
"/web/sales/**", "GET", "hasAnyRole('ROLE_PSCAdmin','ROLE_PSCUser')"
```

**Note:** The method (GET in this case) must be specified in the 11.7 oeablSecurity.csv file. Specifying the method was optional in the 11.6 intercept-url elements in the oeablSecurity-XXXX-XXXX.xml files.

**See Also**

- If you are updating from a previous release, you must migrate your Spring Security configurations to the new properties and CSV files before you can run on 11.7. The process is done after the 11.7 installation, and is automated in the Progress Developer Studio. See *OpenEdge Getting Started: Migrating to OpenEdge 11.7* for more information.

- For information about individual properties, see the oeablSecurity.properties.README file.

- For more configuration information, see *Progress Application Server for OpenEdge: Administration Guide*.
Release 11.7 Documentation

This chapter describes the product documentation provided for OpenEdge Release 11.7.

For details, see the following topics:

- Changes to the documentation set
- Accessing OpenEdge documentation
- Release 11.7 documentation set

Changes to the documentation set

For this release, the following manuals are added to the documentation set:

- OpenEdge Getting Started: Migrating to OpenEdge 11.7
- OpenEdge Getting Started: OpenEdge Authentication Gateway Guide
- OpenEdge Getting Started: Change Data Capture
- OpenEdge Management and OpenEdge Explorer: Managing Change Data Capture in databases

For this release, the following manuals are retired from the documentation set:

- OpenEdge Data Management: DataServer for ODBC
- OpenEdge: Migrating to OpenEdge Business Process Management 11.6
Accessing OpenEdge documentation

You can download the manuals and Web papers that are being delivered for 11.7 from the Progress Software Download Center: http://www.progress.com/support-and-services/evaluation-support/download-resources/download-center. Download files are provided for both Windows and UNIX platforms:

- PROGRESS_OE_11.7_WIN_DOC.zip
- PROGRESS_OE_11.7_UNIX_DOC.tar.gz

Each file contains the documentation in PDF format, development samples and example procedure files, plus instructions for installing them. The installation instructions are in the readme.txt file at the root of the zip file directory structure.

You also can access latest OpenEdge documentation in HTML format from https://documentation.progress.com

You can also access the entire OpenEdge 11.7 product documentation set, in both HTML and PDF format, from the Progress Communities: https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx

Release 11.7 documentation set

The following table lists the 11.7 documentation available for each technology area. You can access the entire OpenEdge 11.7 product documentation set on Progress Communities: https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx.

Table 3: Release 11.7 documentation map arranged by technology

<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABL application development</td>
<td>OpenEdge Getting Started: Guide for New Developers</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Getting Started: ABL Essentials</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: ABL Reference</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Debugging and Troubleshooting</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Error Handling</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: ProDataSets</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td></td>
<td>Web Paper: ABL Database Triggers and Indexes</td>
</tr>
<tr>
<td></td>
<td>Web Paper: ABL Data Types Addenda</td>
</tr>
<tr>
<td></td>
<td>Web Paper: Batch-mode Event Support</td>
</tr>
<tr>
<td></td>
<td>Web Paper: Coding for Portability</td>
</tr>
<tr>
<td></td>
<td>Web Paper: Dynamic Call Object</td>
</tr>
<tr>
<td>To learn about . . .</td>
<td>See . . .</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ABL application development with the OpenEdge GUI for .NET | OpenEdge Getting Started: GUI for .NET Primer  
OpenEdge Development: GUI for .NET Mapping Reference  
OpenEdge Development: GUI for .NET Programming |
| Auditing                                                | OpenEdge Getting Started: Core Business Services - Security and Auditing  
OpenEdge Data Management: Database Administration  
OpenEdge Development: Programming Interfaces |
| Business intelligence and reporting                     | OpenEdge Reporting: Query/Results Administration and Development  
OpenEdge Reporting: Query/Results for UNIX  
OpenEdge Reporting: Query/Results for Windows  
OpenEdge Reporting: Deploying Crystal Reports  
OpenEdge Reporting: Report Builder Deployment |
| Change Data Capture                                     | OpenEdge Getting Started: Change Data Capture  
OpenEdge Management and OpenEdge Explorer: Managing Change Data Capture in databases  
OpenEdge Data Management: Database Administration  
OpenEdge Development: Basic Database Tools  
OpenEdge Data Management: SQL Development  
OpenEdge Data Management: SQL Reference  
OpenEdge Development: Programming Interfaces |
| Data management                                         | OpenEdge Getting Started: Database Essentials  
OpenEdge Data Management: Database Administration  
OpenEdge Data Management: DataServer for Microsoft SQL Server  
OpenEdge Data Management: DataServer for Oracle  
OpenEdge Data Management: SQL Development |
| Deployment                                               | OpenEdge Deployment: Managing ABL Applications  
OpenEdge Deployment: Startup Command and Parameter Reference  
OpenEdge Deployment: WebClient Applications |
<p>| Error handling                                          | OpenEdge Development: Error Handling |</p>
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
</table>
| Identity management | OpenEdge Getting Started: Core Business Services - Security and Auditing  
OpenEdge Getting Started: Identity Management  
OpenEdge Getting Started: Multi-tenancy Overview  
OpenEdge Development: Programming Interfaces |
| Installation and configuration | OpenEdge Getting Started: Installation and Configuration  
OpenEdge Getting Started: Migrating to OpenEdge 11.7  
OpenEdge Deployment: Startup Command and Parameter Reference |
| Internationalization and localization | OpenEdge Development: Internationalizing Applications  
OpenEdge Development: Translation Manager  
OpenEdge Development: Visual Translator |
| Mobile applications | OpenEdge Application Server: Developing AppServer Applications  
OpenEdge Application Server: Administration  
Online Help: Progress Developer Studio for OpenEdge Guide |
| Multi-tenancy | OpenEdge Getting Started: Multi-tenancy Overview  
OpenEdge Getting Started: Identity Management  
OpenEdge Management and OpenEdge Explorer: Getting Started with Multi-tenancy  
OpenEdge Management and OpenEdge Explorer: Configuring Multi-tenancy  
OpenEdge Development: Programming Interfaces |
| Open Client interfaces to a Java or .NET application or a Sonic Native Invocation or SOAP Web service definition | OpenEdge Getting Started: Application and Integration Services  
OpenEdge Development: Open Client Introduction and Programming  
OpenEdge Development: Java Open Clients  
OpenEdge Development: .NET Open Clients  
OpenEdge Development: Web Services  
OpenEdge Development: Messaging and ESB  
Online Help: ProxyGen |
| OpenEdge Business Process Management | OpenEdge Getting Started: Installation and Configuration  
OpenEdge Getting Started: Developing BPM Applications with Developer Studio  
Business Process Server: First Steps Guide |
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
</table>
**Business Process Server: Server Administrator’s Guide**  
**Business Process Server: Application Developer’s Guide**  
**Business Process Server: Customization Guide**  
**Business Process Server: Managed Adapter Guide**  
**Business Process Server: Troubleshooting Guide**  
**OpenEdge Development: ABL Reference**  
**OpenEdge BPM BusinessManager: Clustering Guide**  
**Online Help: Progress OpenEdge BPM Studio User Guide**  
**Online Help: Progress OpenEdge Business Process Server Documentation**  
**Online Help: Progress OpenEdge Business Process Portal User Guide** |
| **OpenEdge Business Rules** | **Online Help: Progress Developer Studio for OpenEdge**  
**Web Paper: Introducing OpenEdge Business Rules**  
**Web Paper: Integrating OpenEdge Business Rules: An end-to-end workflow** |
| **OpenEdge Management and OpenEdge Explorer** | **OpenEdge Management and OpenEdge Explorer: Configuring Multi-tenancy**  
**OpenEdge Management and OpenEdge Explorer: Getting Started with Multi-tenancy**  
**OpenEdge Management and OpenEdge Explorer: Getting Started**  
**OpenEdge Management and OpenEdge Explorer: Configuration**  
**OpenEdge Management and OpenEdge Explorer: Managing Table Partitioning in Databases**  
**OpenEdge Management: Database Management**  
**OpenEdge Management: Resource Monitoring**  
**OpenEdge Management: Servers, DataServers, Messengers, and Adapters**  
**OpenEdge Management: Alerts Guide and Reference** |
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenEdge Replication</td>
<td>OpenEdge Replication: User Guide</td>
</tr>
<tr>
<td>OpenEdge Table Partitioning</td>
<td>OpenEdge Getting Started: Table Partitioning</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Management and OpenEdge Explorer: Managing Table Partitioning in Databases</td>
</tr>
<tr>
<td>OpenEdge Studio</td>
<td>OpenEdge Getting Started: Progress OpenEdge Studio</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: AppBuilder</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Basic Database Tools</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Basic Development Tools</td>
</tr>
<tr>
<td></td>
<td>Web Paper: Application Development Environment (ADE) Addenda</td>
</tr>
<tr>
<td>Progress Application Server for OpenEdge</td>
<td>Progress Application Server for OpenEdge: Introducing PAS for OpenEdge</td>
</tr>
<tr>
<td></td>
<td>Progress Application Server for OpenEdge: Administration Guide</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Management: Progress Application Server for OpenEdge Configuration</td>
</tr>
<tr>
<td></td>
<td>Progress Application Server for OpenEdge: Application Migration and Development Guide</td>
</tr>
<tr>
<td></td>
<td>AppServer Properties Readme</td>
</tr>
<tr>
<td></td>
<td>Catalina Properties Readme</td>
</tr>
<tr>
<td></td>
<td>Openedge Properties Readme</td>
</tr>
<tr>
<td>Progress Developer Studio for OpenEdge</td>
<td>OpenEdge Getting Started: Introducing the Progress Developer Studio for OpenEdge Visual Designer</td>
</tr>
<tr>
<td></td>
<td>Online Help: Progress Developer Studio for OpenEdge Guide</td>
</tr>
<tr>
<td>To learn about . . .</td>
<td>See . . .</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
| Progress Dynamics    | OpenEdge Development: Progress Dynamics Getting Started  
|                      | OpenEdge Development: Progress Dynamics Basic Development  
|                      | OpenEdge Development: Progress Dynamics Administration  
|                      | OpenEdge Development: Progress Dynamics Advanced Development  
|                      | OpenEdge Development: Progress Dynamics Managers API Reference  
|                      | OpenEdge Development: Progress Dynamics Repository Reference  
|                      | OpenEdge Development: Progress Dynamics Web Development Guide  |
| Open Client interfaces to a Java or .NET application or a Sonic Native Invocation or SOAP Web service definition | OpenEdge Getting Started: Application and Integration Services  
|                      | OpenEdge Development: Open Client Introduction and Programming  
|                      | OpenEdge Development: Java Open Clients  
|                      | OpenEdge Development: .NET Open Clients  
|                      | OpenEdge Development: Web Services  
|                      | OpenEdge Development: Messaging and ESB  
|                      | Online Help: ProxyGen  |
| REST Web services    | OpenEdge Development: Web Services  
|                      | OpenEdge Application Server: Developing AppServer Applications  
|                      | OpenEdge Application Server: Administration  
|                      | Online Help: Progress Developer Studio for OpenEdge Guide  |
| Security             | OpenEdge Getting Started: Core Business Services - Security and Auditing  
|                      | OpenEdge Getting Started: OpenEdge Authentication Gateway Guide  
|                      | OpenEdge Getting Started: Identity Management  
|                      | OpenEdge Development: Programming Interfaces  |
| Service oriented architectures (SOA) | OpenEdge Getting Started: Application and Integration Services  
|                      | OpenEdge Application Server: Developing AppServer Applications  
|                      | OpenEdge Application Server: Administration  
|                      | OpenEdge Development: Web Services  
|                      | ubroker.properties.README  |
| SmartObjects and ADM | OpenEdge Development: ADM and SmartObjects  
<p>|                      | OpenEdge Development: ADM Reference  |</p>
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP Web services</td>
<td>OpenEdge Getting Started: Application and Integration Services</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Web Services</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Application Server: Developing AppServer Applications</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Application Server: Administration</td>
</tr>
<tr>
<td></td>
<td>Online Help: ProxyGen</td>
</tr>
<tr>
<td>Sonic messaging and application integration environment</td>
<td>OpenEdge Development: Messaging and ESB</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>OpenEdge Development: Debugging and Troubleshooting</td>
</tr>
<tr>
<td>WebSpeed (Classic)</td>
<td>OpenEdge Getting Started: WebSpeed Essentials</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Application Server: Developing WebSpeed Applications</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
<tr>
<td>WebSpeed on PAS for OpenEdge</td>
<td>Progress Application Server for OpenEdge</td>
</tr>
<tr>
<td></td>
<td>Progress Application Server for OpenEdge: Application Migration and Development Guide</td>
</tr>
<tr>
<td></td>
<td>Progress Application Server for OpenEdge: Administration Guide</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Management: Progress Application Server for OpenEdge Explorer Configuration</td>
</tr>
<tr>
<td>Working with JSON and XML</td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Working with XML</td>
</tr>
</tbody>
</table>