OpenEdge® Getting Started: New and Revised Features
Notices

© 2015 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

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

Business Making Progress, Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Deliver More Than Expected, Easy, Fathom, Icenium, Kendo UI, Making Software Work Together, OpenEdge, Powered by Progress, Progress, Progress Control Tower, Progress RPM, Progress Software Business Making Progress, Progress Software Developers Network, Rollbase, RulesCloud, RulesWorld, SequeLink, Sitefinity (and Design), SpeedScript, Stylus Studio, TeamPulse, Telerik, (Telerik (and Design), Test Studio, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, AppsAlive, AppServer, BravePoint, BusinessEdge, DataDirect Spy, DataDirect SupportLink, , Future Proof, High Performance Integration, Modulus, NativeScript, OpenAccess, Pacific, ProDataSet, Progress Arcade, Progress Pacific, Progress Profiles, Progress Results, Progress RFID, Progress Progress Software, ProVision, PSE Pro, SectorAlliance, Sitefinity, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

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

The Release Notes can be found in the OpenEdge installation directory and online at:

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

October 2015

Last updated with new content: Release 11.6.0
# Table of Contents

## Preface

- Purpose ................................................................. 7
- Audience ............................................................... 8
- Organization ........................................................... 8
- Using this manual ................................................... 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 .................................................. 13
- OpenEdge messages .................................................. 13
  - Obtaining more information about OpenEdge messages .... 14

## Chapter 1: What's New in 11.6

- Installation and platform support ................................ 18
- Startup parameters .................................................... 18
- OpenEdge ABL enhancements ....................................... 19
  - Enumerated types .................................................. 19
  - Reflection ............................................................ 20
  - Serializing classes to binary and JSON formats ............... 22
  - New attributes and startup parameters for parsing XML ...... 22
  - Single-line comments ............................................ 23
  - Additional information from XREF and XREF-XML options of COMPIL... 23
  - RadForm subclasses in GUI for .NET ......................... 23
- OpenEdge RDBMS ..................................................... 24
- OpenEdge Replication ................................................. 25
- Progress Developer Studio for OpenEdge ......................... 25
- WebSpeed support in Pacific Application Server for OpenEdge ............................................................. 27
- Data Object Services and OpenEdge Mobile support ............ 28
- OpenEdge Server Technology ....................................... 29
- OpenEdge Business Process Management ......................... 31
- OpenEdge SQL .......................................................... 32
- OpenEdge DataServers ................................................. 33
- OpenEdge Management and OpenEdge Explorer ................ 34

## Chapter 2: 11.6 Feature Comparisons

OpenEdge Getting Started: New and Revised Features
WebSpeed administration differences and migration issues.................................................................37
Data Object Service terminology and URI differences.............................................................................38

Chapter 3: Release 11.6 Documentation.................................................................41
Changes to the documentation set.................................................................................................................41
Accessing OpenEdge documentation.......................................................................................................41
Release 11.6 documentation set.................................................................................................................42
Preface

For details, see the following topics:

- Purpose
- Audience
- Organization
- Using this manual
- 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 11.6.0, and directs you to where you can find more detailed information about these changes in the documentation set. The Release 11.6 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 11.6 from Release 11.5.

**Organization**

*What's New in 11.6 on page 17*

Presents brief explanations of new and enhanced features introduced in Release 11.6, and references places within the documentation set where more detailed feature information can be found.

*11.6 Feature Comparisons on page 37*

Presents brief explanations of features whose behavior has changed since Release 11.5.

*Release 11.6 Documentation on page 41*

Provides information about the OpenEdge Release 11.6 documentation set.

**Using this manual**

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

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


**References to ABL compiler and run-time features**

ABL is both a compiled and an interpreted language that executes in a run-time engine. The documentation refers to this run-time engine as the **ABL Virtual Machine (AVM)**. When the documentation refers to ABL source code compilation, it specifies ABL or the compiler as the actor that manages compile-time features of the language. When the documentation refers to run-time behavior in an executing ABL program, it specifies the AVM as the actor that manages the specified run-time behavior in the program.
For example, these sentences refer to the ABL compiler’s allowance for parameter passing and the AVM’s possible response to that parameter passing at run time: "ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error." The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: "The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets."

**References to ABL data types**

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

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

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

**Typographical conventions**

This manual uses the following typographical and syntax conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.</td>
</tr>
<tr>
<td><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, GET and CTRL.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>A plus sign between key names indicates a simultaneous key sequence: you press and hold down the first key while pressing the second key. For example, CTRL+X.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>A space between key names indicates a sequential key sequence: you press and release the first key, then press another key. For example, ESCAPE H.</td>
</tr>
</tbody>
</table>
### Convention

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fixed width</strong></td>
<td>A fixed-width font is used in syntax, code examples, system output, and file names.</td>
</tr>
<tr>
<td><strong>Fixed-width italics</strong></td>
<td>Fixed-width italics indicate variables in syntax.</td>
</tr>
<tr>
<td><strong>Fixed-width bold</strong></td>
<td>Fixed-width bold italic indicates variables in syntax with special emphasis.</td>
</tr>
<tr>
<td><strong>UPPERCASE fixed width</strong></td>
<td>ABL keywords in syntax and code examples are almost always shown in uppercase. Although shown in uppercase, you can type ABL keywords in either uppercase or lowercase in a procedure or class.</td>
</tr>
<tr>
<td><strong>Period (.) or colon (:)</strong></td>
<td>All statements except <strong>DO, FOR, FUNCTION, PROCEDURE, and REPEAT</strong> end with a period. <strong>DO, FOR, FUNCTION, PROCEDURE, and REPEAT</strong> statements can end with either a period or a colon.</td>
</tr>
<tr>
<td><strong>[]</strong></td>
<td>Large brackets indicate the items within them are optional.</td>
</tr>
<tr>
<td><strong>[]</strong></td>
<td>Small brackets are part of ABL.</td>
</tr>
<tr>
<td><strong>{}</strong></td>
<td>Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.</td>
</tr>
<tr>
<td><strong>{}</strong></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><strong>...</strong></td>
<td>Ellipses indicate repetition: you can choose one or more of the preceding items.</td>
</tr>
</tbody>
</table>

---

### Examples of syntax descriptions

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

**Syntax**

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

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

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

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

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

**Syntax**

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

In this example, you must specify MESSAGE and at least one expression or SKIP [( 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

This manual 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
- The OpenEdge Product Documentation Overview page on Progress Communities:
  

Once installed, you can locate the example files for this manual in the following path 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\prod\dotnetobjects</td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
<tr>
<td>src\prod\dynamics</td>
<td>The Progress Dynamics documentation</td>
</tr>
<tr>
<td>src\prod\getstartoop</td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td>src\prod\handbook</td>
<td>OpenEdge Getting Started: ABL Essentials</td>
</tr>
<tr>
<td>src\prod\interfaces</td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>src\prod\json</td>
<td>OpenEdge Development: Working with JSON</td>
</tr>
<tr>
<td>src\prod\langref</td>
<td>OpenEdge Development: ABL Reference</td>
</tr>
<tr>
<td>src\prod\prodatasets</td>
<td>OpenEdge Development: ProDataSets</td>
</tr>
<tr>
<td>src\prod\tranman</td>
<td>OpenEdge Development: Translation Manager</td>
</tr>
<tr>
<td>src\prod\visualdesigner</td>
<td>OpenEdge Getting Started: Introducing Progress Developer Studio for OpenEdge Visual Designer</td>
</tr>
<tr>
<td>src\prod\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 \texttt{ENTER}. Details about that message number appear.
4. Press \texttt{F4} to close the message, press \texttt{F3} to access the Procedure Editor menu, and choose \texttt{File > Exit}. 
What's New in 11.6

This chapter summarizes new and enhanced features in OpenEdge® Release 11.6. 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 ABL enhancements
- OpenEdge RDBMS
- OpenEdge Replication
- Progress Developer Studio for OpenEdge
- WebSpeed support in Pacific Application Server for OpenEdge
- Data Object Services and OpenEdge Mobile support
- OpenEdge Server Technology
- OpenEdge Business Process Management
- OpenEdge SQL
- OpenEdge DataServers
- OpenEdge Management and OpenEdge Explorer
Installation and platform support

OpenEdge release 11.6 installation and platform support includes the following:

- **OpenEdge custom installation bundle utility** — You can install all the OpenEdge products using the OpenEdge cdimage or specific products by creating a smaller OpenEdge cdimage (installation bundle) that you can use to install on the target system. The advantages of this installation bundle are that you can copy it to different locations to install the same set of products on multiple computers, or may include it with the application developed by a partner.

  You can create the installation bundle that contains only the files required for the selected products using the OpenEdge custom installation bundle utility. When you run the installation program from the installation bundle that you created, only your selected OpenEdge products are installed.

- **JBOSS Enterprise Edition Installation** — When you install OpenEdge, you can select either of the following:
  - **JBOSS Community Edition** option button to install the JBoss Community Edition server that this installer provides.
  - **JBOSS Enterprise Edition** option button to point OpenEdge to the JBoss Enterprise Edition server that you have installed for your production environment.

- **Java8 Runtime certification** — All OpenEdge supported platforms are certified for Java8 Runtime certification.

- **Infragistics** — OpenEdge release 11.6.0 uses Infragistics NetAdvantage for .NET v2015 Vol 1.CLR 4.0. The Infragistics controls are upgraded to version 15.1.20151.2132.

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 11.6 includes new startup parameters, described in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>For more Information see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>-enforceMm</td>
<td>Use Enforce Mm (-enforceMm) to specify that the behavior of the Message Buffer Size (-Mm) startup parameter follow prior releases.</td>
<td>OpenEdge RDBMS on page 24</td>
</tr>
</tbody>
</table>
For more Information see...

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>For more Information see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>-entityExpansionLimit</td>
<td>Use Entity Expansion Limit (-entityExpansionLimit) to set an upper limit on how many entity substitutions the XML parser (DOM or SAX) will allow in a document.</td>
<td>New attributes and startup parameters for parsing XML on page 22</td>
</tr>
<tr>
<td>-strictEntityResolution</td>
<td>Use Strict Entity Resolution (-strictEntityResolution) to control whether or not the XML parser (DOM or SAX) will attempt to resolve an external entity if that entity is located outside of the directories in the SCHEMA-PATH attribute of any given SAX-reader or X-document object handle or XML-SCHEMA-PATH attribute of the WEB-OBJECT system handle.</td>
<td>New attributes and startup parameters for parsing XML on page 22</td>
</tr>
</tbody>
</table>

For more information, including default values, see:

Manuals: OpenEdge Deployment: Startup Command and Parameter Reference

**OpenEdge ABL enhancements**

OpenEdge Release 11.6 adds support for the ABL features described in the following sections.

- **Enumerated types** on page 19
- **Reflection** on page 20
- **New attributes and startup parameters for parsing XML** on page 22
- **Serializing classes to binary and JSON formats** on page 22
- **Single-line comments** on page 23
- **Additional information from XREF and XREF-XML options of COMPILE** on page 23
- **RadForm subclasses in GUI for .NET** on page 23

**Enumerated types**

ABL includes enumerated types, also known as enums. An enum is made up of a list of strongly typed, named constants called members. The value of a variable defined as an enum is restricted to the list of members defined for that enum. Some of the advantages of enums are that they are self-documenting, strongly typed, and validated at compile time (i.e., the compiler verifies that a variable defined as an enum can only contain values defined by that enum).
You can define two types of enum. In the standard version, a variable defined as an enum can only be assigned one value at a time. You can also define a flag enum. Any variable defined as a flag enum can be assigned any combination of the members (also referred to as “flags” for flag enums) at one time.

All enums implicitly inherit from one of two classes. The single-value enums inherit from Progress.Lang.Enum, and flag enums inherit from Progress.Lang.FlagsEnum, which is a subclass of Progress.Lang.Enum. Both procedural-based and class-based functionality has been added to support enums. For example:

- Users can define their own enums using the ENUM statement and work with them dynamically at run time using the DYNAMIC-ENUM function.

- You can use the new bitwise operators AND, OR, XOR, and NOT to manipulate the state of flag-based enums or use the class-based methods SetFlag(), ToggleFlag(), and UnsetFlag() to achieve the same results.

- Many existing ABL constructs, including the CASE statement and the comparison operators (e.g., NE, EQ), also work with enums. In addition, those ABL constructs can be used with .NET enums. The Progress.Util.EnumHelper class is still supported for backwards compatibility, but most of its functionality can now be achieved directly with ABL statements and functions.

For more information, see:

| Manuals: | OpenEdge Development: ABL Reference |

**Reflection**

In Release 11.6, the reflection capabilities in ABL's object-oriented functionality has been expanded. New methods for Progress.Lang.Class provide information about a class's constructors, events, methods, properties, data members, and implemented interfaces, as summarized in the following table:

**Table 2: New Progress.Lang.Class methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetConstructor()</td>
<td>Returns a Progress.Reflect.Constructor instance (or an array of instances) containing</td>
</tr>
<tr>
<td>GetConstructors()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetEnumName ( )</td>
<td>information on the constructor(s) described by the input parameters.</td>
</tr>
<tr>
<td>GetEnumValues ( )</td>
<td>Provide information about enum instances. See Enumerated types on page 19 for more information about ABL's new support for enumerated types.</td>
</tr>
<tr>
<td>IsEnum ( )</td>
<td>Returns a Progress.Reflect.Event instance (or an array of instances) containing information on the event(s) described by the input parameters.</td>
</tr>
<tr>
<td>IsFlagsEnum ( )</td>
<td>Returns an array of Progress.Lang.Class instances containing information on the interfaces that the class implements.</td>
</tr>
<tr>
<td>GetEvent ( )</td>
<td>Returns a Progress.Reflect.Method instance (or an array of instances) containing information on the method(s) described by the input parameters.</td>
</tr>
<tr>
<td>GetEvents ( )</td>
<td>Returns a Progress.Reflect.Property instance (or an array of instances) containing information on the properties described by the input parameters.</td>
</tr>
<tr>
<td>GetInterfaces ( )</td>
<td>Returns a Progress.Reflect.Variable instance (or an array of instances) containing information on the variable(s) described by the input parameters.</td>
</tr>
<tr>
<td>GetMethod ( )</td>
<td>Returns TRUE if a given .NET class is generic.</td>
</tr>
<tr>
<td>GetMethods ( )</td>
<td>Returns TRUE if the given .NET class is indexed.</td>
</tr>
</tbody>
</table>

Specific details on the parameters and overloads of each of these methods can be found in their entries in OpenEdge Development: ABL Reference.

As mentioned in the table, there are new built-in classes in the Progress.Reflect package that provide information about a class's members (e.g., Progress.Reflect.Method). Several new built-in enumerated types (e.g., Progress.Reflect.AccessMode, Progress.Reflect.Flags) are used as parameters for reflection methods and values of class properties. See Enumerated types on page 19 for an introduction to ABL's support of enums.

For more information, see:

| Manuals: | OpenEdge Development: ABL Reference  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
</tbody>
</table>
Serializing classes to binary and JSON formats

In Release 11.4, ABL added the capability to serialize class-based objects and pass them between an ABL client and an AppServer. In 11.6, you can now serialize objects to binary or JSON format via the Progress.IO.BinarySerializer and Progress.IO.JsonSerializer classes. Two new classes, Progress.IO.InputStream and Progress.IO.OutputStream, provide the default functionality for input and output streams, respectively, and two subclasses, Progress.IO.FileInputStream and Progress.IO.FileOutputStream, facilitate reading objects from and writing objects to a file.

As with passing objects between client and AppServer, the class of any object serialized to binary or JSON must be defined as SERIALIZABLE. By default, binary serialization includes all data members, properties, temp-tables, and ProDataSets, regardless of whether they are public, protected, or private. However, only data members and properties defined as public participate in JSON serialization by default. Protected and private class members must be themselves defined as SERIALIZABLE in the DEFINE VARIABLE, DEFINE PROPERTY, DEFINE TEMP-TABLE, and DEFINE DATASET statements.

For more information, see:

| Manuals:         | OpenEdge Development: ABL Reference |

New attributes and startup parameters for parsing XML

Four new attributes are available to enhance security when parsing XML. For a SAX-reader or X-document object handle:

- ENTITY-EXPANSION-LIMIT: The integer value of this attribute is the maximum number of entity substitutions allowed for an XML document.
- STRICT-ENTITY-RESOLUTION: When this attribute is set to TRUE, the parser will not attempt to resolve an external entity if the entity is located outside of the directories in the SCHEMA-PATH.

For the WEB-CONTEXT system handle:

- XML-ENTITY-EXPANSION-LIMIT: The integer value of this attribute is the maximum number of entity substitutions allowed for an XML document sent as a WebSpeed request.
- XML-STRICT-ENTITY-RESOLUTION: When this attribute is set to TRUE, the parser will not attempt to resolve an external entity if the entity is located outside of the directories in the XML-SCHEMA-PATH attribute for an XML document sent as a WebSpeed request.

These behaviors can also be controlled globally using the -entityExpansionLimit and -strictEntityResolution startup parameters.

<table>
<thead>
<tr>
<th>Manuals:</th>
<th>OpenEdge Development: ABL Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OpenEdge Deployment: Startup Command and Parameter Reference</td>
</tr>
</tbody>
</table>
Single-line comments

In addition to the existing comment style, bracketed by /* and */, ABL now supports single-line comments. Single-line comments are preceded by // and end at a newline character, carriage return, or end of file. To be properly interpreted as a single-line comment, the // must be preceded by a line break or whitespace.

Caution: If you need to use // at the beginning of a pathname, such as in `RUN //dir1/file`, the pathname will be interpreted as a comment because // is preceded by a space. Enclosing the pathname in quotes ensures that the text following // will not be treated as a comment. Pathnames that contain // in their interior, such as in `RUN parent//dir1/file`, will work as expected.

For more information, see:

| Manuals: | OpenEdge Development: ABL Reference |

Additional information from XREF and XREF-XML options of COMPILE

In Release 11.6, the XREF and XREF-XML options of the COMPILE statement display SEARCH information for static queries on local temp-tables within a procedure.

For more information, see:

| Manuals: | OpenEdge Development: ABL Reference |

RadForm subclasses in GUI for .NET

 Telerik's UI for WinForms includes three form classes: RadForm, RadRibbonForm, and ShapedForm. To properly support these forms in GUI for .NET applications, we implemented subclasses that extend the Telerik form classes in the same way we have extended the standard System.Windows.Forms.Form class with Progress.Windows.Form. The new subclasses are:

- Progress.Windows.OERadForm
- Progress.Windows.OERadRibbonForm
- Progress.Windows.OEShapedForm
- Progress.Windows.MDIChildRadForm
- Progress.Windows.MDIChildRadRibbonForm
- Progress.Windows.MDIChildShapedForm

These can be accessed through the form creation wizard in Progress Developer Studio for OpenEdge.
OpenEdge RDBMS

OpenEdge Release 11.6 includes the following new features for the OpenEdge RDBMS:

- **Partition Copy utility**
  The partition copy utility allows a database administrator to recover accidentally deleted data from a partition without taking the database offline. To copy a partition from a temporary database, the PROUTILITY utility is enhanced with the PARTITIONMANAGE COPY qualifier. The PROUTILITY utility DBRESTRICT is also enhanced to protect your database during partition copy operations.

- **LOB support in PROUTILITY DBANALYS**
  The PROUTILITY database analysis utilities are enhanced to include analysis of LOB data. When LOB data is found in an area, a “LOB Summary” follows the “Record Block Summary” in the output, and LOB data is included in the combined summary. LOB data is also written to a new *.lob.txt file when the –csoutput parameter is used on the analysis command line.

- **Network message buffers (–Mm)**
  Network message buffer sizes, specified by the –Mm startup parameter, are no longer required to agree between client and server. The client value is used as a suggestion during startup, but the value specified by the server is used once the connection is made.

  For backward compatibility, you can require that the client and server values of –Mm agree by specifying –enforceMm.

- **Virtual System Table changes**
  The following fields are added to the _Connect Virtual System Table:
  - _Connect-PartitionId — Displays the partition ID if the database is enabled for Table Partitioning and the wait type is REC or RGET.
  - _Connect-Timestamp — Displays the timestamp of the user’s schema cache.
  - _Connect-Wait2 — Displays additional info based on the value of _Connect-Wait1. If _Connect-Wait1 contains a ROWID, this field contains a table number. If _Connect-Wait1 contains a dbkey, this field contains an area number.

  **Note:** There are additional VST changes supporting OpenEdge Replication. For details, see OpenEdge Replication on page 25.

  Add these fields to existing VSTs in your database with PROUTILITY UPDATEVSTS.

For more information on these topics, see:

<table>
<thead>
<tr>
<th>Manual:</th>
<th>OpenEdge Data Management: Database Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online help:</td>
<td>Database Administration Utilities</td>
</tr>
</tbody>
</table>
OpenEdge Replication

OpenEdge Release 11.6 includes the following new features for OpenEdge Replication:

- **Agent Restart**
  Replication agents can be restarted without restarting the target database server. The DSRUTIL RESTART command accepts “agent” as a valid qualifier. Prior to this release, the only way to restart an agent was to stop and restart the target database server.

- **Virtual System Table changes**
  New VSTs are added, and existing VSTs are enhanced, to fully provide all the information previously available only in the Replication Monitor tool.
  
  The following existing tables have additional fields:
  
  - `_ReplAgent`
  - `_ReplAgentControl`
  - `_ReplServer`

  The following tables are added:
  
  - **_ReplAgentActivity** — Contains dynamic information about active OpenEdge Replication agents
  - **_ReplAgentControlActivity** — Contains dynamic information about the active OpenEdge Replication agents an OpenEdge Replication server is controlling.
  - **_DbServiceManager** — Contains detailed information about an OpenEdge Replication service manager.
  - **_DbServiceManagerObjects** — Contains detailed information about registered objects of an OpenEdge Replication service manager.

  Add these fields to existing VSTs in your database with PROUTIL UPDATEVSTS.

For more information on these topics, see:

| Manual: | OpenEdge Replication: User Guide |

Progress Developer Studio for OpenEdge

OpenEdge Release 11.6 contains Progress Developer Studio for OpenEdge updates to provide the following features:

- **WebSpeed support in PAS for OpenEdge** — Pacific Application Server for OpenEdge (PAS for OpenEdge) is available as a host for WebSpeed applications. You can migrate existing WebSpeed applications to PAS for OpenEdge, or you can use the Progress Developer Studio for OpenEdge to develop new WebSpeed applications that run on PAS for OpenEdge.
You can create an OpenEdge project with the ABL Web App project type to create and deploy one or more ABL services (for WebSpeed, REST, or Data Object service types) as a single Web application to PAS for OpenEdge.

The `web-disp.p` control program available for classic WebSpeed (using the WebSpeed Transaction Server) does not exist in PAS for OpenEdge. On PAS for OpenEdge, Web object execution is controlled by a built-in handler object. This default handler class can be modified to implement any `web-disp.p` customizations that you want to replicate. However, you cannot migrate `web-disp.p` to a PAS for OpenEdge instance.

**OpenEdge Data Object changes** — OpenEdge Mobile services are rebranded as OpenEdge Data Object Services. Like Mobile services in previous releases, Data Object Services support mobile app development and also can be used to provide OpenEdge data to web apps, which can run in any browser-based environment.

However, the OpenEdge Mobile App Builder is no longer supported. You can now use the Telerik Platform to create mobile apps. For more information, see Data Object Services and OpenEdge Mobile support on page 28.

In addition, the Mobile OpenEdge project type is replaced with two new OpenEdge project types:

- **Data Object** — To create and deploy Data Object Services to classic OpenEdge servers
- **ABL Web App** — To create and deploy Data Object Services to PAS for OpenEdge

**Changes to OpenEdge project types** — To create Data Object Services for deployment to classic OpenEdge servers, you create a Data Object project. For PAS for OpenEdge, you can create an ABL Web App project for the following service types:

- **WebSpeed (WebHandler)** — To create and deploy WebSpeed services
- **REST (Mapped RPC)** — To create and deploy REST services
- **Data Object (Annotated RPC)** — To create and deploy Data Object Services

**Profiler Editor** — Progress Developer Studio provides an built-in profiler editor. You can use the Profiler tab in Run Configuration to profile a session. You can view the details of the profiled session in the editor.

**References Update to Telerik assemblies** — You can update references to Telerik assemblies if you have upgraded to a new release of Progress Developer Studio for OpenEdge, and you want to update all references to work with the new release.

Progress Developer Studio for OpenEdge updates the assemblies automatically the first time you open a workspace. However, if you decline to perform the automatic update, later when you launch the Update Assembly References tool from the Progress Developer Studio for OpenEdge menu, all instances of `assemblies.xml` and `toolbox.xml` in your current workspace automatically appear in the list of files to be updated, and all of the necessary settings are in place by default.

**Telerik UI for WinForms** — Progress Developer Studio for OpenEdge installs a trial version of Telerik UI for WinForms by default. It includes controls that you can use to build your business applications. If you need a different version of Telerik, you can download it from the Telerik web site.

The Toolbox in the Visual Designer Editor includes all the Telerik UI for WinForms controls.

**Assert API in ABLUnit** — The Assert API provides a set of assertion methods to validate the expected result of a method or procedure in a test with that of the actual result. This API is a
WebSpeed support in Pacific Application Server for OpenEdge

Beginning in OpenEdge Release 11.6, Pacific Application Server (PAS) for OpenEdge is available as a host for WebSpeed applications. You can migrate existing WebSpeed applications to PAS for OpenEdge, or you can use the Progress Developer Studio for OpenEdge to develop new WebSpeed applications that run on PAS for OpenEdge. In addition, OpenEdge Management and OpenEdge Explorer are extended to provide configuration and management support for WebSpeed on PAS for OpenEdge.

Some of the advantages of WebSpeed on PAS for OpenEdge compared to classic WebSpeed are:

- WebSpeed on PASOE employs a more integrated architecture compared to classic WebSpeed because both the Web server and the WebSpeed Transaction sever are combined in a single instance.
- WebSpeed on PASOE is more efficient than classic WebSpeed regarding the management and the availability of the agents that handle client requests.
- Classic WebSpeed only supported the GET and POST HTTP verbs. WebSpeed on PASOE supports all standard HTTP verbs.
- WebSpeed on PASOE supports event procedures that were not supported on classic WebSpeed.
- PAS for OpenEdge includes support for multiple servers in a single instance; you do not need to configure and run separate Web server, WebSpeed Transaction server, and AppServer instances.
- PAS for OpenEdge shares a single security context among the WEB transport that supports WebSpeed and the other transports (REST, SOAP, and APSV).
- Security issues that are inherent in CGI-compatible Web servers for classic WebSpeed are absent from the Pacific Application Server.
- Application deployment in WebSpeed on PAS for OpenEdge is simplified because all files (static, dynamic, executable) can be combined in a single package.
- Classic WebSpeed only supports a limited set of file types for uploading. WebSpeed on PAS for OpenEdge supports all media types for uploading and downloading. You can now, for example, stream binary files, PDFs, etc.
- If you are creating new WebSpeed applications, you can take advantage of an ABL-based OpenHTTP object oriented layer for Web application development (similar to Java servlet APIs).

Infragistics controls upgrade — The Infragistics controls are upgraded to version 15.1.20151.2132.

For more information on these topics, see:

| Online help: | Progress Developer Studio for OpenEdge |

• Classic WebSpeed does not have access to a number of server attributes of the SESSION system handle. PAS for OpenEdge can access the following attributes:
  • SERVER-CONNECTION-BOUND
  • SERVER-CONNECTION-BOUND-REQUEST
  • SERVER-CONNECTION-CONTEXT
  • SERVER-CONNECTION-ID
  • SERVER-OPERATING-MODE

The topic, WebSpeed administration differences and migration issues on page 37 contains more information about classic WebSpeed compared to PAS for OpenEdge.

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pacific Application Server for OpenEdge: Introducing PAS for OpenEdge</td>
</tr>
<tr>
<td>Pacific Application Server for OpenEdge: Application Migration and Development Guide</td>
</tr>
<tr>
<td>Pacific Application Server for OpenEdge: Administration Guide</td>
</tr>
<tr>
<td>OpenEdge Management: Pacific Application Server for OpenEdge Configuration</td>
</tr>
</tbody>
</table>

Data Object Services and OpenEdge Mobile support

OpenEdge Release 11.6 contains the following updates to mobile support:

• OpenEdge Mobile retired and replaced with OpenEdge Data Object Services — As previously noted (see Progress Developer Studio for OpenEdge on page 25), OpenEdge Mobile services have been rebranded and reconfigured as OpenEdge Data Object Services. Similar to Mobile services in previous releases, Data Object Services support mobile and web app access to OpenEdge data using server-side Data Object resources implemented as ABL Business Entities.

However, the Mobile App Builder is no longer supported and OpenEdge itself provides no other embedded support for building mobile apps. You can use the Telerik Platform, or your own choice of JavaScript tools, to build mobile or web apps with access to Data Object Services using client-side JavaScript Data Objects (JSDOs). The JSDO itself is also available through a Progress open source project.

The open source Cloud Data Object project on GitHub (see below) offers the JSDO support previously available in OpenEdge. This JSDO support is provided with the latest release of Progress Data Objects, which is the Progress-supported implementation of the Cloud Data Object.

The libraries provided with the Progress Data Objects implementation are built into the Progress Data Service template of the Telerik AppBuilder. You can also obtain these libraries from the GitHub Cloud Data Object project itself for use with your own choice of JavaScript development tools.

JSDO documentation previously available with OpenEdge is also updated and available as part of the Cloud Data Object project on GitHub. Note that the previously released documentation for the JSDO, OpenEdge Development: Mobile Applications, has been retired from the OpenEdge documentation set.
You can build server-side Data Objects and Services using the Progress Developer Studio for OpenEdge. In addition to the online help provided with Progress Developer Studio for OpenEdge, an overview of OpenEdge Data Object Services and detailed information on coding ABL Business Entities as Data Objects is provided in *OpenEdge Development: Web Services*.

- **Submit operation enhancement** — The signature of the ABL routine that implements the Submit operation for a Data Object resource previously supported only an OpenEdge ProDataSet parameter for a ProDataSet resource that supports before-imaging. With this release, you can instead use a single temp-table parameter. However, a Submit operation on single temp-table parameter, while sending multiple record changes per network request, does not support before-imaging. Therefore, you must provide your own custom implementation to manage the record changes that are sent in a temp-table parameter with Submit.

- **Terminology and URI changes** — With the rebranding of Mobile services as Data Object Services, a number of changes have been made to terminology in the Data Object documentation. In addition, the default location for certain client files generated by OpenEdge for deployment in Web applications has changed in order to simplify the URIs for accessing these files over the Web. For more information, see Data Object Service terminology and URI differences on page 38.

For more information on Progress support for using client-side Data Objects (JSDOs) and OpenEdge Data Object Services, see:

<table>
<thead>
<tr>
<th>Web:</th>
<th>Cloud Data Object open source project on GitHub: <a href="http://clouddataobject.github.io/">http://clouddataobject.github.io/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Progress Data Objects Guide and Reference:</em> <a href="http://documentation.progress.com/output/pdo/">http://documentation.progress.com/output/pdo/</a>*</td>
</tr>
<tr>
<td>Manual:</td>
<td><em>OpenEdge Development: Web Services</em></td>
</tr>
<tr>
<td>Online help:</td>
<td><em>Progress Developer Studio for OpenEdge</em></td>
</tr>
</tbody>
</table>

### OpenEdge Server Technology

OpenEdge Release 11.6 contains Server Technology updates to provide the following features:

- **SSL security updates** —
  - OpenSSL Library is upgraded to 1.0.1m.
  - OpenEdge supports TLSv1.1 and TLSv1.2 protocols that also secure applications against the Padding Oracle On Downgraded Legacy Encryption (POODLE) vulnerability.
  - OpenEdge supports digital certificate validation whose signatures use SHA2 (256 or 512) message digest algorithms.

For more information on compatibility between clients and servers of different versions and a complete list of supported protocols, ciphers, and certificates, see *OpenEdge Getting Started: Core Business Services - Security and Auditing*.

- **Application Server Refresh** — The Application Server (AppServer) Refresh enhancement allows you to trim and restart individual agents gradually to allow new agents to replace older agents. Use the `-refresh` command for `ASBMAN` or `WTBMAN` to perform the refresh action.
The refresh operation tries to maintain run-time capacity equal to or greater than what it was at the time refresh started. At the end of the refresh, the broker adds new agents only if the current number is less than when the refresh started. Progress Software Corporation recommends that you configure your maximum server setting to a higher number as the number of agents may increase during the App Server Refresh. This enhancement can help you perform the following actions:

- Reload r-code
- Trim file system allocation
- Recover process memory

**Caution:** The refresh action may take a considerable time to finish. If you are using the App Server Refresh feature to load new r-code, be aware that clients that were connected to the App Server before the refresh action was performed have two versions of the r-code between the CONNECT and DISCONNECT calls. All clients that connect after the refresh is performed get new agents and run new r-code.

For more information on these topics, see:

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

Progress OpenEdge release 11.6 provides the following OpenEdge BPM features and improvements:

- **ProDataSet and TempTable enhancements** — The following enhancements are made to ProDataSet and TempTable dataslots:

  - The BPM portal provides the capability to visualize ProDataSet and TempTable dataslots, and supports CRUD operations for these dataslots. Users can Add, Edit and Delete the records either using ProDataset or TempTable. While adding or editing a record, users can define how relations between different TempTables are handled. Users can see the relation information while performing Create or Edit actions.

- OpenEdge BPM supports two new Query APIs, `insertRow` and `deleteRow`, for DataSet and TempTable. These APIs can be accessed from the JS Tools and the GEL Tools.

- **Configurable Menu Items in Portal** — Home tab menus can be enabled from the Business Process Portal Configuration tab during User/Group creation. Users can customize visibility of menu options under Home tab based on user setting.

- **JBoss EAP Installer support** — With the integration of Jboss Enterprise Edition Server, users can choose between installing the Business Process Server either on the community edition or on the enterprise edition server using the OpenEdge installer.

- **SBMServerDCS in OEBPM** — SBMServerDCS support is provided to enable connections between ABL API's and an SSL enabled BPServer.

- **JavaDocs API** — Java docs APIs are shipped with the OE BPM product and can be found in `%DLC%\oebpm\server\docs`.

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>Progress OpenEdge Business Process Modeler: User’s Guide</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Getting Started: Developing BPM Applications with Developer Studio</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Web Paper: Migrating to OpenEdge 11.6</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 SQL

OpenEdge Release 11.6 contains OpenEdge SQL updates to provide the following features:

• **Enhanced ICU Collations support** — OpenEdge provides support for newer linguistic collations from the libraries that are part of ICU 4.8. This is in addition to the ICU 2.4 libraries that are currently supported. The prefix ICU_48- identifies collations from the newer ICU libraries shipped with OpenEdge (ICU 4.8).

• **SQL Performance optimization**
  
  • **GROUP BY optimization** — With the introduction of GROUP BY optimization, STREAM aggregation is performed for GROUP BY columns. There are two types of GROUP BY optimization:
    
    • **Re-order GROUP BY columns** — Stream aggregation is performed for GROUP-BY column queries that are the leading prefix components of an index.
    
    • **Considering eqi-constant predicates** — Using the Equi-Constant Predicates optimization technique, leading index prefix components with eqi-constant predicates are identified, and an index scan is performed if the remaining index components are specified in group-by columns.

• **UPDATE STATISTICS** — Enhancements provide OpenEdge SQL users with increased adaptability, quicker development, and in some cases better scalability. After creating new SQL statistics with UPDATE STATISTICS, better scalability arises from SQL adapting to changing table sizes and configurations automatically.

• **ORDER BY, OFFSET FETCH, and TOP in sub-queries** — The ORDER BY clause and the OFFSET FETCH clause can now be used in a SQL sub-query. The TOP phrase can now also be used in a SQL sub-query. Prior to this release, these were limited to the top-level query. Support for these in sub-queries helps further refining of the selection of records in a query and improving performance. The clauses for ORDER BY and OFFSET FETCH can be combined in a SQL query when selecting records from a data set. The clauses ORDER BY and OFFSET FETCH have a relationship with the TOP phrase.

• **SELECT DISTINCT** — The OpenEdge SQL query SELECT DISTINCT is optimized in this release. OpenEdge SQL now uses a different, more scalable, and better performing execution model for DISTINCT. In addition, there are related optimizations that can help performance of eligible queries. Queries now utilize the OpenEdge SQL Grouping execution engine for DISTINCT, which uses hash/stream aggregation. These optimizations remove any need for “hand-tuning” of DISTINCT queries, such as by “forcing” index hints. In addition, this release also provides new optimizations that support using applicable indexes to get the distinct valued output rows for a query. This also can improve the performance of this type of query.

• **Autonomous Schema Update** — Autonomous Schema Update (ASU) helps resolve character width constraints by updating the SQL column width in schema automatically when Authorized Data Truncation (ADT) occurs for data in a column. The first time a SQL query is executed with ASU and ADT enabled, truncated data is returned and the column width is adjusted to accommodate complete data. Users should execute the query again to get complete data. The ASU feature is turned off by default. To enable ASU, include the new server startup parameter -SQLWidthUpdate.
OpenEdge DataServers

Progress OpenEdge release 11.6 provides the following OpenEdge DataServers features and improvements:

- **SSL Support**—With this feature, DataServer users have the option of configuring the server component process to require Secure Sockets Layer (SSL) remote client connections. You can maintain both SSL-enabled and non-SSL client sessions by configuring either ProBroker or a DataServer instance of the Unified Broker. Broker instances support either secure or non-secure connections. The OpenEdge client responds to an SSL-enabled server without any specific startup options.

- **Support for DataServer in PAS for OpenEdge environment**—With this release, DataServers users (MS SQL Server and Oracle) can connect to the multi-threaded PAS for OpenEdge AppServer environment and execute ABL. To run ABL in multi-threaded AppServer context, a self-service DataServer must be used to connect to PAS for OpenEdge. If the server component of DataServer is distributed, when you connect to PAS for OpenEdge, the DataServer client runs multi-threaded, while the server runs as a stand alone process using either a ProBroker configuration or a DataServer instance of the Unified Broker.

  **Note:** ODBC DataServer is not supported in PAS for OpenEdge environment.

- **ROWID area restructuring**—OpenEdge ROWID restructuring, introduced in the DataServer for MS SQL Server, is improved by introducing a new variable-length ROWID format. The area stored in the record buffer for ROWID is no longer a fixed size and is not dependent on the ZPRGRS_RECID_BUF_SIZE -Dsrv option. Reduced RECID size causes the DataServer to use smaller network bandwidth while passing records between client and server. This can improve network performance. While the DataServer runtime uses the new ROWID algorithm by default, it allows the old format ROWID (OpenEdge 11.4 and before) to be maintained.

- **New wire-protocol driver support**—The DataServer for MS SQL Server has replaced the Legacy Wire Protocol Driver for MS SQL Server. The new Data Direct driver has improved performance when compared to the legacy driver. The DataServer for MS SQL Server supports the following drivers:
  - MSS 32-bit Wire Protocol Driver for MS SQL Server
  - MSS 64-bit Wire Protocol Driver for MS SQL Server

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>

For more information, see:

<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>

OpenEdge Getting Started: New and Revised Features 33
OpenEdge Management and OpenEdge Explorer

OpenEdge release 11.6 provides the following OpenEdge Management and OpenEdge Explorer features and improvements:

**Dashboard improvements** – The Dashboard has been re-designed to offer improved usability with the following features:

- A new menu is available to quickly navigate to the sub-menus.
- Hover menus have been removed to ensure smoother navigation on touch-screen devices such as tablets.
- Collection views are displayed as tabs to access, configure, and organize views easily.
- An **Adminserver status** viewlet is included in the **Default** collection view to display the number of running and offline AdminServers.
- A **Help Me** option, in the console footer bar, has been provided to access context-sensitive help related to the active console page, and OpenEdge Management and OpenEdge Explorer documentation.

**New graphs** – Graphs in OpenEdge Management are now designed to be more interactive and provide the following features:

- The new graphs belong to the Historical family which show multiple points in time, and their interpretation is in the form of line chart, area chart, column chart, stacked area chart, or stacked column chart.
- All graphs, except viewlets, have three options at the top-right corner for customization. Using these options you can select a period, ranging from 2 hours to 2 weeks, to display the data collected for that period, reload the graph, and select different forms of graphical interpretation.
- The legend of each graph defines different colors of the graph, and shows maximum, minimum, average, and last graph values for selected time period.
- By default, 15 days of graph data is stored for each database resource.

**Pacific Application Server for OpenEdge (PAS for OpenEdge) improvements** – OpenEdge Management and OpenEdge Explorer provides the following improved features for PAS for OpenEdge:

- Supports multiple ABL applications per PAS instance.
- Provides metrics for ABL applications, REST, SOAP, APSV, and WEB transport services.
- Supports configuration of APSV transport properties.
- WEB transport support for PAS for OpenEdge – PAS for OpenEdge now supports deployment of WebSpeed applications and allows you to:
  - Manage WEB transport services.
  - View WEB transport metrics.
  - Configure WEB transport properties.
  - Manage WEB handlers.
- Supports validation of configuration properties to ensure a successful connection.
New OpenEdge database views – A database’s home page provides two new database views:

• **Database configuration** view in **Informational Views** – Displays general information such as details, features, and other parameters of your database.

  **Note:** The **Database Configuration** view is a replacement of the **General Details** view. It does not allow you to modify your database configuration properties.

• **Performance Summary** view in **Operational Views** – Provides summarized information about your database and all other operational views associated with it.

  Use the clickable links on the page to find historical data for particular information for 1 hour, 1 day, 1 week, and 4 weeks.

**Display of alerts** – Alert features provide the following improvements:

• All alerts are now displayed in a grid frame of the **Alerts** page.

• Five of the most recent alerts are displayed on a resource’s home page and its monitoring summary page. When there are more than 5 alerts, use the **See complete list** option to open the **Alerts** page and see all the alerts associated with the resource.

**Other improvements** include:

• The **User Activity** view in **Operational Views** of a database’s home page has been re-designed to be more functional and display the database user management pages in different tabs.

• A **Database connection home** link has been provided in a database’s home page to access the **Database Administration Console** page and manage the database.

For more information, see:

<table>
<thead>
<tr>
<th>Manuals:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenEdge Management and OpenEdge Explorer: Getting Started</td>
</tr>
<tr>
<td>OpenEdge Management: Pacific Application Server for OpenEdge Configuration</td>
</tr>
<tr>
<td>OpenEdge Management: Database Management</td>
</tr>
<tr>
<td>OpenEdge Management: Resource Monitoring</td>
</tr>
<tr>
<td>OpenEdge Management: Alerts Guide and Reference</td>
</tr>
</tbody>
</table>
11.6 Feature Comparisons

This chapter provides comparisons of how features in Release 11.6 differ from previous releases. For details, see the following topics:

- WebSpeed administration differences and migration issues
- Data Object Service terminology and URI differences

WebSpeed administration differences and migration issues

This section describes some of the differences between classic WebSpeed and WebSpeed on PAS for OpenEdge that affect administration and migration of existing applications.

Note: Classic WebSpeed (which is a WebSpeed Transaction Server that uses a third-party Web server) was not changed or updated in OpenEdge Release 11.6. WebSpeed on PAS for OpenEdge is, in many ways, a new and improved platform for WebSpeed application development and deployment but classic WebSpeed is still supported and available.

Administration

Both classic WebSpeed and WebSpeed on PAS for OpenEdge can be managed with OpenEdge Management and OpenEdge Explorer tools. However, note these administrative differences with regard to configuration files and tools:

- WebSpeed Workshop is not supported for WebSpeed on PAS for OpenEdge.
The classic WebSpeed utilities WSCONFIG and WTBMAN are not supported. Use the TCMAN utility for configuring and managing WebSpeed on PAS for OpenEdge.

Classic WebSpeed's configuration properties are specified in a `ubroker.properites` file. On PAS for OpenEdge, WebSpeed configuration properties are specified in an instance's `/conf/openedge.properties` file.

**Migration**

You should be able to migrate existing WebSpeed applications from classic WebSpeed to PAS for OpenEdge with few, if any, changes. However, these are some of the issues that you should take into account before migrating:

- PAS for OpenEdge supports the three most common types of WebSpeed Web objects: static HTML, Embedded SpeedScript, and CGI Wrapper. But, it does not support HTML Mapped Web objects.

- The ABL `DISPLAY` statement is not supported for WebSpeed applications running on PAS for OpenEdge.

- In classic WebSpeed, applications ran under a WebSpeed Transaction Server, which employs a control program, `web-disp.p`, that runs on all agents and executes Web objects. The `web-disp.p` control program does not exist in PAS for OpenEdge. Web object execution is controlled by a built-in handler object. This default handler can be modified to implement any `web-disp.p` customizations that you want to replicate. However, you cannot migrate `web-disp.p` to a PAS for OpenEdge instance.

  Also note that PAS for OpenEdge allows you to create multiple handlers, and to configure each handler with its own URL namespace. In classic WebSpeed, there can only be a single `web-disp.p` and a single URL namespace.

  The following new classes were added to the ABL to support creating handlers for WebSpeed and other HTTP Web applications:

  - `OpenEdge.Web.WebHandler` class
  - `OpenEdge.Web.WebResponseWriter` class
  - `OpenEdge.Web.WebResponse` class
  - `OpenEdge.Web.WebRequest` class
  - `OpenEdge.Web.WebSpeedHandler` class

  See *OpenEdge Development: ABL Reference* for more information about these classes.

- PAS for OpenEdge does not support stateful applications in the same way they are supported in classic WebSpeed. In classic WebSpeed, a WSEU cookie instructs WebSpeed to lock an agent process to a specific client.

  Instead of locking an agent process, PAS for OpenEdge maintains a stateful application by using JSESSIONID cookies. They are passed between the client and the server, and refer to session data stored on the server.

**Data Object Service terminology and URI differences**

The following changes to documentation terminology and URIs have been made to support Data Object Services in this OpenEdge release:
In addition to rebranding OpenEdge Mobile services as OpenEdge Data Object Services, the following table describes associated changes in terminology:

<table>
<thead>
<tr>
<th>This OpenEdge Mobile service term...</th>
<th>Changes to this Data Object Service term or terms...</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>built-in</code> CRUD or submit operation</td>
<td><code>standard</code> CRUD or Submit operation</td>
</tr>
<tr>
<td>Business Entity</td>
<td><code>Business Entity</code>, <code>Data Object resource implementation</code></td>
</tr>
<tr>
<td>JSDO</td>
<td><code>client-side Data Object, JSDO</code></td>
</tr>
<tr>
<td>JSDO catalog</td>
<td><code>Data Service Catalog</code></td>
</tr>
<tr>
<td>Mobile interface</td>
<td><code>Business Entity interface</code>, <code>Data Object resource interface</code></td>
</tr>
<tr>
<td>Mobile operation</td>
<td><code>Data Object operation, resource operation</code></td>
</tr>
<tr>
<td>Mobile resource</td>
<td><code>server-side Data Object, Data Object resource, resource</code></td>
</tr>
<tr>
<td>Mobile service resource</td>
<td><code>Data Object Service resource</code></td>
</tr>
<tr>
<td>Mobile Web application</td>
<td><code>Data Object Web application, Web application</code></td>
</tr>
<tr>
<td><code>non-built-in</code> invoke operation</td>
<td><code>custom</code> Invoke operation</td>
</tr>
<tr>
<td>user login session</td>
<td><code>JSDO login session</code></td>
</tr>
</tbody>
</table>

Default URI change for Data Service Catalog and other client files — In previous releases, the default location in OpenEdge-generated Web applications for deploying JSDO catalog (now Data Service Catalog) and other client files was the `/static/mobile/` subfolder of the Web application folder on the Web server. With this release, the `/mobile/` subfolder has been removed and the default location for these files is now `/static/`. This means that the default absolute URI, for example, to access a deployed Data Service Catalog from the client now takes the following form:

```
scheme://host:port/web-app-name/static/service-name.json
```
Release 11.6 Documentation

This chapter describes the product documentation provided for OpenEdge Release 11.6. For details, see the following topics:

- Changes to the documentation set
- Accessing OpenEdge documentation
- Release 11.6 documentation set

Changes to the documentation set

For this release, the following manual has been retired from the documentation set:

- OpenEdge Development: Mobile Applications

Accessing OpenEdge documentation

You can download the manuals and Web papers that are being delivered for 11.6 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_11.6_WIN_DOC.zip
• PROGRESS_OE_11.6_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.

For your convenience, you can also access the entire OpenEdge 11.6 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 11.6 documentation set

The following table lists the 11.6 documentation available for each technology area. You can access the entire OpenEdge 11.6 product documentation set on Progress Communities: https://community.progress.com/technicalusers/w/openedgegeneral/1329.openedge-product-documentation-overview.aspx.

Table 4: Release 11.6 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>ABL application development</td>
<td>OpenEdge Getting Started: ABL Essentials</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: ABL Reference</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: Debugging and Troubleshooting</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: Error Handling</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: Object-oriented Programming</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: ProDataSets</td>
</tr>
<tr>
<td>ABL application development</td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>ABL application development</td>
<td>Web Paper: ABL Database Triggers and Indexes</td>
</tr>
<tr>
<td>ABL application development</td>
<td>Web Paper: ABL Data Types Addenda</td>
</tr>
<tr>
<td>ABL application development</td>
<td>Web Paper: Batch-mode Event Support</td>
</tr>
<tr>
<td>ABL application development</td>
<td>Web Paper: Coding for Portability</td>
</tr>
<tr>
<td>ABL application development</td>
<td>Web Paper: Dynamic Call Object</td>
</tr>
<tr>
<td>ABL application development with the OpenEdge GUI for .NET</td>
<td>OpenEdge Getting Started: GUI for .NET Primer</td>
</tr>
<tr>
<td>ABL application development with the OpenEdge GUI for .NET</td>
<td>OpenEdge Development: GUI for .NET Mapping Reference</td>
</tr>
<tr>
<td>ABL application development with the OpenEdge GUI for .NET</td>
<td>OpenEdge Development: GUI for .NET Programming</td>
</tr>
<tr>
<td>Auditing</td>
<td>OpenEdge Getting Started: Core Business Services - Security and Auditing</td>
</tr>
<tr>
<td>Auditing</td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td>Auditing</td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>To learn about . . .</td>
<td>See . . .</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Business intelligence and reporting</td>
<td>OpenEdge Reporting: Query/Results Administration and Development</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Reporting: Query/Results for UNIX</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Reporting: Query/Results for Windows</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Reporting: Deploying Crystal Reports</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Reporting: Report Builder Deployment</td>
</tr>
<tr>
<td>Data management</td>
<td>OpenEdge Getting Started: Database Essentials</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Getting Started: Multi-tenancy Overview</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: Database Administration</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: DataServer for Microsoft SQL Server</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: DataServer for ODBC</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: DataServer for Oracle</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Data Management: SQL Development</td>
</tr>
<tr>
<td>Deployment</td>
<td>OpenEdge Deployment: Managing ABL Applications</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Deployment: Startup Command and Parameter Reference</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Deployment: WebClient Applications</td>
</tr>
<tr>
<td>Error handling</td>
<td>OpenEdge Development: Error Handling</td>
</tr>
<tr>
<td>Identity management</td>
<td>OpenEdge Getting Started: Core Business Services - Security and Auditing</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Getting Started: Identity Management</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Getting Started: Multi-tenancy Overview</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Programming Interfaces</td>
</tr>
<tr>
<td>Installation and configuration</td>
<td>OpenEdge Getting Started: Installation and Configuration</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Startup Command and Parameter Reference</td>
</tr>
<tr>
<td>Internationalization and localization</td>
<td>OpenEdge Development: Internationalizing Applications</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Translation Manager</td>
</tr>
<tr>
<td></td>
<td>OpenEdge Development: Visual Translator</td>
</tr>
<tr>
<td>To learn about . . .</td>
<td>See . . .</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| 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  
|                      | Business Process Server: Terminology Guide  
|                      | 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  
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
</table>
|                      | *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*  
|                      | *OpenEdge Management: Trend Database Guide and Reference*  
|                      | *OpenEdge Management: Reporting*  
|                      | *ubroker.properties.README*  
| OpenEdge Replication | *OpenEdge Replication: User Guide*  

---

OpenEdge Getting Started: New and Revised Features  
45
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenEdge Table Partitioning</td>
<td><strong>OpenEdge Getting Started: Table Partitioning</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Data Management: Database Administration</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Management and OpenEdge Explorer: Managing Table Partitioning in Databases</strong></td>
</tr>
<tr>
<td>OpenEdge Studio</td>
<td><strong>OpenEdge Getting Started: Progress OpenEdge Studio</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: AppBuilder</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Basic Database Tools</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Basic Development Tools</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Web Paper: Application Development Environment (ADE) Addenda</strong></td>
</tr>
<tr>
<td>Pacific Application Server for OpenEdge</td>
<td><strong>Pacific Application Server for OpenEdge: Introducing PAS for OpenEdge</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Pacific Application Server for OpenEdge: Administration Guide</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Management: Pacific Application Server for OpenEdge Configuration</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Pacific Application Server for OpenEdge: Application Migration and Development Guide</strong></td>
</tr>
<tr>
<td></td>
<td><strong>AppServer Properties Readme</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Catalina Properties Readme</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Openedge Properties Readme</strong></td>
</tr>
<tr>
<td>Progress Developer Studio for OpenEdge</td>
<td><strong>OpenEdge Getting Started: Introducing the Progress Developer Studio for OpenEdge Visual Designer</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Online Help: Progress Developer Studio for OpenEdge Guide</strong></td>
</tr>
<tr>
<td>Progress Dynamics</td>
<td><strong>OpenEdge Development: Progress Dynamics Getting Started</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Basic Development</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Administration</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Advanced Development</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Managers API Reference</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Repository Reference</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Progress Dynamics Web Development Guide</strong></td>
</tr>
<tr>
<td>To learn about . . .</td>
<td>See . . .</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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: 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**  
**OpenEdge Development: ADM Reference**                                                                        |
| SOAP Web services                                        | **OpenEdge Getting Started: Application and Integration Services**  
**OpenEdge Development: Web Services**  
**OpenEdge Application Server: Developing AppServer Applications**  
**OpenEdge Application Server: Administration**  
Online Help: ProxyGen                                                                                           |
<table>
<thead>
<tr>
<th>To learn about . . .</th>
<th>See . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonic messaging and application</td>
<td><strong>OpenEdge Development: Messaging and ESB</strong></td>
</tr>
<tr>
<td>integration environment</td>
<td></td>
</tr>
<tr>
<td>Troubleshooting</td>
<td><strong>OpenEdge Development: Debugging and Troubleshooting</strong></td>
</tr>
<tr>
<td>WebSpeed (Classic)</td>
<td><strong>OpenEdge Getting Started: WebSpeed Essentials</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Application Server: Developing WebSpeed Applications</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Working with JSON</strong></td>
</tr>
<tr>
<td>WebSpeed on PAS for OpenEdge</td>
<td><strong>Pacific Application Server for OpenEdge: Introducing PAS for OpenEdge</strong></td>
</tr>
<tr>
<td></td>
<td>**Pacific Application Server for OpenEdge: Application Migration and</td>
</tr>
<tr>
<td></td>
<td>Development Guide</td>
</tr>
<tr>
<td></td>
<td><strong>Pacific Application Server for OpenEdge: Administration Guide</strong></td>
</tr>
<tr>
<td></td>
<td>**OpenEdge Management: Pacific Application Server for OpenEdge Explorer</td>
</tr>
<tr>
<td></td>
<td>Configuration**</td>
</tr>
<tr>
<td>Working with JSON and XML</td>
<td><strong>OpenEdge Development: Working with JSON</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OpenEdge Development: Working with XML</strong></td>
</tr>
</tbody>
</table>