Progress®



What's New in OpenEdge 12

OpenEdge®

Copyright

© 2021 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.

DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Icenium, Inspec, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), SpeedScript, Stylus Studio, Stylized Design (Arrow/3D Box logo), Styleized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Habitat, Chef WorkStation, Corticon.js, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Everywhere, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, JustAssembly, JustDecompile, JustMock, KendoReact, NativeScript Sidekick, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Insight, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient 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.

July 2021

Last updated with new content: Release 12.4

Updated: 2021/07/14

Table of Contents

Learn about OpenEdge 12.....11

What's New in OpenEdge 12.4.....13

DevOps Productivity.....14
- Methods as Callbacks.....14
- VAR Statement Enhancements.....14
- New ABL AGGREGATE statement for server-side processing.....16
- Query expression evaluation behavior change.....16
- .NET Core support.....17
- ABL service deployment.....17
- Third-party Tools.....17
- AppBuilder reinstatement.....17

Continuous Operations.....18
- Database maintenance mode.....18
- Default storage areas.....18
- Automatic database reconnection with OpenEdge Replication.....18
- Simplified online table truncation.....18
- Before-image space management.....19
- Drop a field online.....19
- Cancel SQL queries via JDBC clients.....19
- Improved online schema support for delta.df in Data Dictionary.....19
- Default area assignments for database objects (SQL, ABL, Data Dictionary).....20

Security Updates.....21
- TLS support for OpenEdge Replication.....21
- TLS Client Authentication from an ABL Socket.....21
- Online Transparent Data Encryption database policy management.....21

Supportability.....22
- Graceful emergency shutdown of a PAS for OpenEdge multi-session agent.....22
- Enforced Replication synchronization during database shutdown.....22

OpenEdge Command Center.....22

What's New in OpenEdge 12.3.....25

Continuous operations.....26
- Automatic database reconnection.....26
- Simplified Database upgrade.....26
- Uninterrupted port scanning.....26
- Before-image space management.....26
- Reorganize fragmented record data.....27

On-demand login broker changes....27
Relaxed CRC rule for r-code....27
Simplified license changes....27
Change ABL application schema without restart....28
Developer experience....28
ABL enhancements....28
Java and .NET Open Clients....31
SECPROP utility....32
Third-party tools....32
Performance....32
Enhanced server-side join processing....32
Improved database query operations....33
Enhanced index repositioning....33
PROUTIL enhancement for LOB statistics....33
Savepoint support....33
FOR FIRST support....33
Security....34
Remote STS client key management....34
Installation support for client key management....34
OpenEdge security domain concepts....35
OpenEdge SQL....35
Protrace extension on Windows....35
Generate core dumps for SQL servers....35
Pro2 6.2 New Features....36
Database migration....36
CDC enhancements....36
Bulk-load enhancements....36
Release notes....36

What's New in OpenEdge 12.2....37

Continuous operations....38
Performance improvements....41
Security updates....42
OpenSSL, TLS, and Cipher Suite support....43
Developer experience enhancements....43
Updates to ABLUnit....46
Support for deploying an OpenEdge database in a Docker container....48
Java support....49
OpenEdge Database....49
PAS for OpenEdge....49
OpenEdge Adapters....51
OpenEdge Management....52
OpenEdge SQL....52
OpenEdge Pro2 6.1....53

De-supported and deprecated features in OpenEdge 12.2....54

What's New in OpenEdge 12.1....55

High availability....56
Performance....57
Security....58
Developer experience....59
Miscellaneous....60

What's New in OpenEdge 12.0....61

Progress Application Server for OpenEdge....63
HealthScanner....63
Docker Container support....64
Unified logging for PAS for OpenEdge container....64
Swagger UI support for PASOE management APIs....64
Improved agent stop....64
Server-side ABL performance profiling....64
Security updates....65
SQL....65
Online schema changes....66
Online JVM configuration....66
Troubleshoot information added to Protrace report....66
Autonomous Update Statistics....66
Database....66
AI file streaming....67
Multi-threaded server....67
Reduced locking delays in buffer hash table....67
Modifiable VSTs in replication target....67
Appropriate file permissions for conversion utilities....68
Sequences all 64-bit....68
Database startup parameter changes....68
Large file support....68
Synchronous commit mode no longer supported....68
Improved transition control....68
Encryption changes for binary dump and load....69
OpenEdge Management....69
Simplified PAS for OpenEdge creation wizard....69
Simplified PAS for OpenEdge configuration....69
OpenEdge Management now runs independently of AdminServer....69
ABL....70
New memory stream classes....70
Indeterminate array resizing improvement....70
Error and stop object enhancements for a Java open client....70

STOP condition processing....71
Behavior change for empty FIELDS list....71
Client-principal updates....72
New server-side join processing....72
R-code updates....72
Progress Developer Studio for OpenEdge....73
Performance profiler enhancements....73
New ABL editor options....73
Support for importing and exporting workspace settings....73
Bundled third-party tool updates....74
Telerik UI for WinForms update....74
DataServers....74
Requirement to build executables on AIX is lifted....74
Dynamic loading of Oracle client library in OpenEdge DataServer is enabled....74

Preface

Purpose

This book briefly describes both new features and changes to existing features introduced in OpenEdge Release 12, and directs you to the product documentation, where you can find more detailed information about these changes.

Audience

This book is primarily intended for OpenEdge application developers and administrators who are upgrading their license to Release 12.

Organization

What's New in OpenEdge 12.4 on page 13

Describes the new features and other updates delivered in OpenEdge Release 12.4.

What's New in OpenEdge 12.3 on page 25

Describes the new features and other updates delivered in OpenEdge Release 12.3.

What's New in OpenEdge 12.2 on page 37

Describes the new features and other updates delivered in OpenEdge Release 12.2.

What's New in OpenEdge 12.1 on page 55

Describes the new features and other updates delivered in OpenEdge Release 12.1.

What's New in OpenEdge 12.0 on page 61

Describes the new features and other updates delivered in OpenEdge Release 12.0.

Documentation conventions

See Documentation Conventions for an explanation of the terminology, format, and typographical conventions used throughout the OpenEdge content library.

1

Learn about OpenEdge 12

OpenEdge 12 is the most secure, available, and performant version to date. All businesses can benefit from the significant advances in this release. OpenEdge continuously evolves with new features and enhancements to help you meet the demand for application performance, availability, security, and data management. To assist in modernizing your OpenEdge application, this release provides features that:

- Achieve continuous planned and unplanned uptime
- Deliver on performance even as the number of users and connections increases
- Extend capabilities and data across various connections and devices
- Protect from breaches and address compliance mandates
- Create intuitive experiences users demand
- Deliver on business demand-quickly and productively

For details about the features in OpenEdge 12, see:

- What's New in OpenEdge 12.4 on page 13
- What's New in OpenEdge 12.3 on page 25
- What's New in OpenEdge 12.2 on page 37
- What's New in OpenEdge 12.1 on page 55
- What's New in OpenEdge 12.0 on page 61

Customer Validation Program

The OpenEdge Customer Validation Program (CVP) is a collaboration platform for Progress developers and users of OpenEdge to drive overall product improvement. By becoming an OpenEdge CVP member, you will gain access to:

- Roadmap surveys—Your responses to the short roadmap surveys influence product direction and help prioritize deliverables.
- Virtual open houses—Attend interactive webinars to learn about new development and architectural runways.
- Usability surveys—Tell us how you use OpenEdge so that we can improve your experience with it.
- Sprint reviews—See how our product development is moving forward to make sure your needs are being addressed.
- Pre-release software—Try our software before it's released.

The CVP is hosted on Progress Community, where you can share your ideas and network with thousands of OpenEdge developers.

Progress Information Hub

Release 12 of OpenEdge introduces the Progress Information Hub: an all-new self-service portal to the rich set of technical content that is available for OpenEdge, including educational videos, user and reference information, courseware, and more.

Highlights include:

- **Persona-based content**—User assistance is now available in a much more audience-aware presentation, giving you the right breadth of topics that are neatly tailored to the context in which you're using OpenEdge products.
- **Content categories**—The topics most relevant to the tasks you perform, as well as links to new and advanced product features, are surfaced in logically grouped and easily browsed content categories.
- **Modern content services platform**—The Progress Information Hub delivers content that is much more easily searched and is designed to get you to the information you need more quickly. It includes links to social media as well as a mechanism for delivering feedback.
- **Guided journeys**—Technical content is surfaced as a series of maps that efficiently guide you through common tasks that OpenEdge users perform. All guided journeys provide the optimal way to leverage the best practices in developing and deploying OpenEdge applications, configuring and administering the OpenEdge environment, and building a robust, secure, performant, and highly-available environment.

Visit the Progress Information Hub at https://docs.progress.com.

2

What's New in OpenEdge 12.4



OpenEdge Release 12.4 delivers updates and enhancements in the following product areas:

- DevOps Productivity on page 14
- Continuous Operations on page 18
- Security Updates on page 21
- Supportability on page 22
- OpenEdge Command Center on page 22

Important: New Relic plugins for PAS for OpenEdge and the OpenEdge Database.

As of June 2021, OpenEdge discontinued support for the NewRelic plugins for PAS for OpenEdge and the OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to the view plugin dashboard. This limitation is a result of the dropped support by New Relic of the OpenEdge plugins as of Wednesday, June 16th, 2021. The official announcement is available from https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

For details, see the following topics:

- DevOps Productivity
- Continuous Operations
- Security Updates

- Supportability
- OpenEdge Command Center

DevOps Productivity

Methods as Callbacks

ABL SAX-READER object

ABL developers can now use ABL object-oriented class methods as a callback target for the OpenEdge SAX-READER object. Before this release, only procedures could be used as callback target for the SAX-READER object. Developers can now use an object-oriented solution for handling callback events within their application and avoid mixing procedural and object-oriented code. Overloaded methods can also be specified as callback handlers.

For more information, see the the following topics:

- 'SAX callback reference' in *Use XML with ABL Applications*
- 'ABL SAX callbacks' in *Use XML with ABL applications*
- 'EVENT-HANDLER-OBJECT' attribute in *ABL Reference*

SOAP web services

ABL developers can now use class methods as SOAP header callback targets for SOAP web service operations. Before this release, only procedures could be used as callback target to handle SOAP service callbacks. ABL developers can now register class methods as event handlers to use built-in ABL functionality to run SOAP web service operations asynchronously. This feature allows overloaded methods in ABL classes to be specified as callback handlers.

For more information, see the the following topics:

- 'SET-CALLBACK( ) method' in *ABL Reference*
- 'Statements to consume SOAP web services' in *Develop an ABL SOAP Client*
- 'Attributes, methods, and events to consume web services' in *Develop an ABL SOAP Client*
- 'Read an XML file using methods as callback' in *Develop an ABL SOAP Client*

VAR Statement Enhancements

Support for expressions in VAR statement

Expressions are now supported during variable initialization using the VAR statement. Expressions can be used to initialize a scalar variable as well as parameters to a constructor when initializing a class variable. The ability to use expressions simplifies your code, allowing you to combine operations into a single statement.

The following example shows defining and initializing scalar variables using expressions:

```
VAR INT a = 50, b = 20, x = a + b , y = a - b, z = x - y.
```

The following example shows defining and initializing variables with a built-in function, a user-defined function, and a method as the expression:

```
VAR DATETIME dtm = DATETIME(TODAY,MTIME).
VAR INTEGER x = myFunction().
VAR INTEGER y = classMethod(output v).
```

The following example shows defining determinate and indeterminate array variables and initializing them using expressions:

```
VAR INT[2] w = [123, a + b].
VAR INT[3] x = [funct( ), 123, a + b].

VAR INT[ ] y = [123, a + b].
VAR INT[ ] z = [funct( ), 123, a + b].
```

The following example shows expressions used as parameters to a constructor:

```
VAR INTEGER x = 10, y = 20.
VAR point p1 = NEW point (x - 5, y + 5).

VAR point p2 = NEW point (classMethod() - offset, someobject:getylocation()).
```

For more information, see the 'VAR statement' in the *ABL Reference* or Learn about the VAR statement' in *Develop ABL Applications*.

Support for instantiation and initialization of a class using the VAR statement

In this release of OpenEdge, you can now instantiate a class when you declare a class variable with the VAR statement by using the NEW function. This feature simplifies your code and allows you to do both operations in a single statement.

```
VAR myclass myobj = NEW myclass("Progress", 2021, ?).
```

For more information, see the 'VAR statement' in the *ABL Reference* or Learn about the VAR statement' in *Develop ABL Applications*.

Support for array object variable instantiation and initialization using the VAR statement

In this release, you can now create an array of objects using the VAR statement. This feature simplifies your code, allowing you to combine multiple operations into a single statement.

The following example shows the instantiation and initialization of a determinate array object variable:

```
VAR StateClass[2] objArrayA = [NEW StateClass(“MA”), NEW StateClass(“NH”)].
```

The following example shows the instantiation and initialization of an indeterminate array object variable:

```
VAR StateClass[] objArrayB = [NEW StateClass(“MA”), NEW StateClass(“NH”)].
```

For more information, see the 'VAR statement' in the *ABL Reference* or Learn about the VAR statement' in *Develop ABL Applications*.

New ABL AGGREGATE statement for server-side processing

This release introduces a new ABL AGGREGATE statement for computing aggregate values on the database server-side. Previously, aggregate calculations were performed client-side using both the ACCUMULATE statement and the ACCUM function. Now, you can simply use a single AGGREGATE statement for the COUNT, TOTAL, and AVERAGE aggregate operations. The AGGREGATE statement simplifies your code and improves performance as the computation is done efficiently on the database server-side.

The following example uses the AGGREGATE statement to calculate the average balance for users in Chicago:

```
VAR DECIMAL avgBalance.

AGGREGATE avgBalance = AVERAGE(Balance) FOR Customer
  WHERE Country EQ 'USA' AND City EQ 'Chicago'.

MESSAGE "Average balance: " avgBalance
  VIEW-AS ALERT-BOX.
```

For more information, see the AGGREGATE statement reference entry in the *ABL Reference*.

Query expression evaluation behavior change

In this release, the AVM no longer ignores errors that occur when evaluating a query expression. Previously, if an error occurred, the value of the expression was set to the Unknown (?) value, and the query would continue to execute. This behavior affects all statements that involve query WHERE clauses, such as FOR EACH statements, FIND statements, and OPEN-QUERY statements. For a dynamic query, the error may not be detected until the first GET-FIRST/NEXT/PREV/LAST method after the query is opened. It is possible that mistakes in existing ABL will be uncovered by this change. Now, any error that occurs is raised and handled by the normal error handling mechanisms.

Here is an unrealistic, but illustrative, example:

```
FOR EACH customer WHERE custNum > INTEGER("A"):
...
END.
```

This code will now raise an error.

Previously, the code resolved to the following:

```
FOR EACH customer WHERE custNum > ?:
...
END.
```

To catch the error raised, use a CATCH block in the block containing the FOR EACH statement (not inside the FOR EACH block as that block has not actually started yet).

If using this feature causes any issues, you can use the -zNoQryExpErrors startup parameter to revert to the old behavior. Please note that this is meant to be used as a temporary work-around. It is possible that by using the startup parameter, you may occasionally return incorrect data from the query. Progress Software strongly recommends that in the long-term, you fix your code to avoid any new errors rather than use this undocumented startup parameter.

.NET Core support

Support for .NET Core has been added to OpenEdge Open Client Toolkit. Open Client for .NET supports .NET framework 4.8, .NET Core 3.1, and .NET Core 5.0. Open Client applications can be run on .NET platforms that support .NET standard 2.0 targeted libraries. Progress Software also supports running Open Client application on Linux .NET core 3.1, Windows .NET core 3.1, and .NET framework 4.8 on Windows.

For details, see 'Runtime support with .NET Core FAQ' in *OpenEdge Frequently Asked Questions*.

ABL service deployment

An ABL service is a set of business logic that can be accessed through a URL path.

ABL Service Deployment creates a `.oeds` file that allows you to:

- Package an ABL service into one archive file
- Easily deploy your service to a PAS for OpenEdge instance
- Build a custom service quickly
- Easily maintain your service and benefit from CI/CD processes

For details, see 'ABL Service Deployment' in *Manage Progress Application Server for OpenEdge*.

Third-party Tools

In OpenEdge 12.4, the following third-party tools that are bundled with Progress Developer Studio for OpenEdge were updated. For more information about all third-party tools packaged with OpenEdge, see the OpenEdge Product Availability Guide.

Infragistics controls update

The "OE Ultra Controls .NET" license ships "Infragistics Ultimate UI for Windows Forms". For OpenEdge 12.4, the version was updated to 19.2.20192.476.

AppBuilder reinstatement

Starting in OpenEdge release 12.0, Progress Software stopped supporting 32-bit Windows versions of OpenEdge products, including AppBuilder. AppBuilder is a multi-purpose application development environment. It can be used as a visual programming environment to create applications. This release includes 32-bit support for AppBuilder.

For more information, see 'Learn About AppBuilder' in *Progress Developer Studio for OpenEdge Online Help*.

Continuous Operations

Database maintenance mode

Certain administrative tasks being performed on an active database are complicated by data changing while the task is underway, due to continuing inflight transactions and new logins. Data administration activities like mass load, purge, synchronizing replication source to target or making a large schema change can be easier to perform with data change paused. The solution is to temporarily restrict user activity and perform maintenance tasks without taking the database down.

This release introduces a restricted online mode that enables you to execute maintenance tasks without user activity and with all the buffer pool and helper processes still available, which improves the performance for these tasks. When this mode is enabled only database utilities can connect to the database.

For details, see 'Restrict database access' in *Manage the OpenEdge Database*.

Default storage areas

You can configure a default Type II storage area for the creation of database objects by object type, and change default areas at any time, whether or not the database is running. This eliminates downtime when moving tables, LOBs, and indexes from a Type I to a Type II storage area.

Note:

For an overview of storage areas, see 'Storage areas' in *OpenEdge Database Essentials*.

In previous releases, when no area was specified, objects were created in a Type I area. To take advantage of performance features, you had to stop the database and move objects to a Type II area.

For more information, see 'Change default storage areas for created objects while the database is running' in *Optimize the OpenEdge Database for Continuous Operation*.

Automatic database reconnection with OpenEdge Replication

OpenEdge provides a mechanism for clients to automatically connect to an OpenEdge Replication target database, if the connection to the replication source database fails. In this release, PAS for OpenEdge agents can now reconnect to the new replication source when `-autoReconnect` is used.

For more information, see 'Automatic database reconnection with OpenEdge Replication' in *Manage ABL Applications*.

Simplified online table truncation

Now you can quickly and efficiently truncate all the data associated with a particular table residing in Type II areas. The data storage space that was occupied by the table is returned to the storage area for reuse by other objects.

In this release, the PROUTIL utility lets you truncate, and optionally move, a table while the database is running, without having to recreate the schema.

For more information, see 'Truncate tables' in *Manage the OpenEdge Database*.

Before-image space management

Space management for the before-image (BI) files was introduced in OpenEdge 12.3 with a mechanism that scans the BI file clusters for committed transactions and recyclable clusters. In OpenEdge 12.3, this was executed as part of the watchdog process. In this release, it is moved from the watchdog process to its own before-image manager process, `probim`. To run BI space management in this release, you must set `-biscantime` to a value greater than zero and run the before-image manager (BIM).

To start the before-image manager, enter:

```
probim <db-name>
```

For more information, see 'PROBIM command' in *Manage the OpenEdge Database*.

Drop a field online

You can now remove a named field from the schema definition while still being connected to the database by means of the online `Drop Field` transaction. The data of the field in the database table is also effectively removed in the process without requiring any exclusive lock.

For more information, see 'Drop a field online' in *Optimize the OpenEdge Database for Continuous Operation*.

Cancel SQL queries via JDBC clients

You can now use the `JDBC Cancel() API` to terminate a SQL query in a JDBC client. When the JDBC client executes the API, the OpenEdge SQL server cancels the query and discards all running operations for that query without impacting other queries or applications. The `JDBC Cancel() API` can be used to cancel `SQL SELECT` queries that are taking longer than expected.

Fore more information, see 'Cancel SQL Queries via JDBC' in *OpenEdge 12.4 Release Enablement*.

Improved online schema support for delta.df in Data Dictionary

The Data Administration tool connects the two databases, compares the schemas, and generates a `.df` file that contains the changes required in order for the first database schema to match the second database schema. This is referred to as the `delta.df`.

Four sections are added to the new `delta.df` file. The sections explain the operations that are performed before and after the deployment of an application:

Sections	Description
PreDeploy	This section contains all non-disruptive online operations and items such as: • Allow the field data type to change from **Integer** to **Int64**. • Modify the existing non-structural schema fields. • Rename a field in an existing table.
Trigger	This section contains all the trigger definition changes such as add, delete, and modify. The database triggers have r-code and are deployed before the trigger schema changes. The trigger r-code is executed after the PreDeploy changes are made.
PostDeploy	This section contains all online disruptive operations. These schema change operations are applied to the production system after confirming that the objects are not used in any other version of the running application.
Offline	This section contains all schema change operations. These schema change operations cannot be carried out online and therefore require the following exclusive schema locks on the database: • Drop an inactive index from an existing table. • Drop a non-indexed field from an existing table.

For more information, see 'Dump incremental delta.df supporting online schema changes' in *OpenEdge Database Tools*.

Default area assignments for database objects (SQL, ABL, Data Dictionary)

By default, ABL and SQL create new data objects in the schema area, and the Data Dictionary creates new database objects in the next storage area after the schema area. To take advantage of the performance capabilities of Type II areas, you can specify a default area for the creation of tables, LOBs, and indexes, so that they are created in a Type II area instead of a Type I area like the schema area. You may also change the default object creation area at any time, whether or not the database is running. Default area assignments eliminate downtime when moving tables, LOBs, and indexes from a Type I to a Type II storage area.

For more information, see 'Set default storage areas for creating database objects' in *Manage the OpenEdge Database*.

You can now use the PROUTIL utility to configure or display default areas.

For more information, see 'PROUTIL SETOBJECTDEFAULTAREA qualifier' in *Manage the OpenEdge Database*.

To create objects in default areas, you can use the Index Editor option, Modify Table option or Field Editor Add option in the Data Dictionary.

For more information, see 'Schema Menu Reference' in *OpenEdge Database Tools*.

The Data Dictionary displays the default area settings for database objects in the Database Properties screen and the Utilities menu's Information option.

For more information, see 'Database Properties' in *OpenEdge Database Tools*.

Security Updates

TLS support for OpenEdge Replication

OpenEdge Replication now supports Transport Layer Security (TLS) for network communication between the source and target databases. This provides OpenSSL encryption for data on the move: data replication between the Replication Server and Replication agent, the Replication Transition process, and coordination of the replication processes.

This is independent of and complementary to OpenEdge Transparent Data Encryption (TDE) which is for data at rest.

For more information, see 'Transport Layer Security' in *Manage the OpenEdge Database*.

TLS Client Authentication from an ABL Socket

OpenEdge now supports client-side TLS authentication from an ABL client socket, in addition to the server-side authentication that was available in earlier OpenEdge versions. The OpenEdge system or security administrator can configure TLS client authentication from an ABL client socket, for a TLS-enabled connection, to any HTTPS server that requires TLS client authentication. Here HTTPS server includes PAS for OpenEdge instances and non-OpenEdge servers such as Apache HTTP Server and Microsoft IIS. This feature enables you to meet the enterprise security policy requirements.

The `CONNECT( )` method of the `Socket` object connects a socket to the specified TCP/IP port on the specified host.

The following socket connection parameters are updated for this feature.

- `-SSLAuth` *`Authentication-type`*
- `-SSLKeyFile` *`filename`* | *`target-database`*
- `-SSLKeyPwd` *`password`*

For more information, see 'CONNECT( ) method (Socket object)' in *ABL Reference*.

Online Transparent Data Encryption database policy management

OpenEdge databases that use Transparent Data Encryption (TDE) have an encryption policy for the database master key (DMK) known as the encryption DB policy, for short. You can perform TDE DB policy management tasks on your TDE-enabled databases while the database is running.

Although you should change object policies regularly for data protection, there are three use cases when you should change the encryption DB policy:

- If you deploy your product with a TDE-enabled template database, change the encryption DB policy so that each installation will have a unique set of object encryption keys.
- If your enterprise security policies need larger master keys, or keys with a different algorithm type, change the encryption DB policy.
- If you think that backup copies of both the database and the keystore have been compromised, change the encryption DB policy as insurance.

If you need to change the encryption DB policy for one of the preceding reasons, you can also change the cipher for certain database objects at the same time.

For more information, see 'Online TDE DB policy management' in *Manage the OpenEdge Database*.

Note: For replication-enabled databases, the source database must be OpenEdge 12.4 or higher to use this feature.

Supportability

Graceful emergency shutdown of a PAS for OpenEdge multi-session agent

An emergency shutdown of a multi-session agent is initiated when a severe error occurs. The shutdown process causes the multi-session agent to redirect all new requests to a new multi-session agent and records a message in the `syslog` (UNIX) or `eventlogs` (Windows) and generates a Protrace file. The shutdown process also creates a new file, `pasoe-agent-crash-trace-`*`process-id`*, in the `$DLC/WRK/`*`InstanceName`*`/work` directory.

The default wait time for the shutdown process is 10 minutes but it can be customized as required.

For more information, see 'Graceful emergency shutdown of a multi-session agent' in *Manage Progress Application Server (PAS) for OpenEdge*.

Enforced Replication synchronization during database shutdown

In this release, the PROSHUT `-replSync` option offers a better way to enforce a guaranteed synchronization of all databases without any pending transactions after a broker shutdown.

For more information, see 'Enforce source and target database synchronization during shutdown' in *Use Database Replication*.

OpenEdge Command Center

Overview

The OpenEdge Command Center is a cloud-ready OpenEdge management console capable of managing multiple OpenEdge resources and installations across various machines and versions of OpenEdge. OpenEdge Command Center is optimized for managing complex, high scale, hybrid cloud OpenEdge deployments.

The following are some of the features introduced in OpenEdge Command Center 1.0.

Highly available OpenEdge Command Center server

The server powers the web console and APIs of OpenEdge Command Center. It uses a MongoDB database and a file store to store data. You can deploy the server as single node or, if configured for high availability, a multi-node cluster.

For more information, see 'Install OpenEdge Command Center in high availability mode' in *Use the OpenEdge Command Center*.

Lightweight Command Center agents

Command Center agents are lightweight processes that are co-located with the OpenEdge installations of the resources you want to manage. You can use a single agent to manage multiple OpenEdge installations on a computer.

For more information, see 'Learn about OpenEdge Command Center' in *Use the OpenEdge Command Center*.

Support for tablet and iPad devices

OpenEdge Command Center provides a modern web console that is developed using latest UI and UX technologies. Administrators can access the console using modern web browsers on their desktops, tablets, or iPad devices.

Self-service dashboard

The dashboard is the console home page of OpenEdge Command Center. It provides resources, such as Getting Started video, banner notifications for "To Do" tasks, quick links, and so on, that helps administrators manage their tasks efficiently.

For more information, see 'OpenEdge Command Center console dashboard' in *Use the OpenEdge Command Center*.

TLS support for OpenEdge Command Center components

To establish a secure network channel for communication between the OpenEdge Command Center server, agents, and configuration database, you can enable TLS.

For more information, see 'Set up TLS' in *Use the OpenEdge Command Center*.

Built-in user management

Built-in user management ensures that administrators have secure control of OpenEdge Command Center across all instances. OpenEdge Command Center provides the following user roles:

- Super administrator
- Administrator
- No access

For more information, see 'Add a new user' in *Use the OpenEdge Command Center*.

PAS for OpenEdge management

Administrators can use the intuitive UI of OpenEdge Command Center to manage PAS for OpenEdge instances in their environment and perform these tasks:

- Discover new PAS for OpenEdge instances in the environment
- Create, start, stop, and delete PAS for OpenEdge instances
- Configure various properties of PAS for OpenEdge instances
- View ABL applications and manage ABL application deployment

For more information, see 'Configure and manage PAS for OpenEdge instances' in *Use the OpenEdge Command Center*.

Open standard REST APIs with Swagger UI documentation

To support customer business needs for application integration and access, OpenEdge Command Center uses open standard REST APIs accompanied with interactive Swagger UI documentation.

For more information, see *OpenEdge Command Center REST APIs*.

3

What's New in OpenEdge 12.3



OpenEdge Release 12.3 delivers updates and enhancements in the following product areas:

- Continuous operations
- Developer experience
- Performance
- Security updates
- OpenEdge SQL updates
- Pro2 6.2 New Features on page 36

Important: New Relic plugins for PAS for OpenEdge and the OpenEdge Database.

As of June 2021, OpenEdge discontinued support for the NewRelic plugins for PAS for OpenEdge and the OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to the view plugin dashboard. This limitation is a result of the dropped support by New Relic of the OpenEdge plugins as of Wednesday, June 16th, 2021. The official announcement is available from https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

For details, see the following topics:

- Continuous operations
- Developer experience

- Performance
- Security
- OpenEdge SQL
- Pro2 6.2 New Features
- Release notes

Continuous operations

Automatic database reconnection

OpenEdge now provides a mechanism for clients to automatically connect to an OpenEdge Replication primary target database, if the connection to the replication source database fails. This mechanism uses the `-autoReconnect` client session startup parameter, and is available when using a client/server connection.

For more information, see "Automatic database reconnection with OpenEdge Replication" in *Manage ABL Applications*.

Simplified Database upgrade

This release enhances the utility that converts replication-enabled OpenEdge 11 databases to OpenEdge 12. New options on the `PROUTIL conv1112` command let you back up and restore only the schema area. You can use the enhanced utility to convert replication-enabled databases from OpenEdge Release 11.*x* to OpenEdge Release 12.3 without requiring a rebase of the target databases.

The `replsource` option backs up physical changes in the schema area of the Release 11 replication source database and the `repltarget` option restores the schema area to the Release 12 replication target database.

For details, see "Convert replication-enabled databases" in *Upgrade to OpenEdge 12*.

Here is a video that shows how to upgrade replication-enabled databases to OpenEdge 12 without a rebase:

Simplified Database Upgrade With OpenEdge 12

Uninterrupted port scanning

You can now use popular port scanning software with OpenEdge Replication without interruptions during database transitions or database log files filled with invalid network messages. By shielding transitions from port scanners or malicious attacks, this enhancement improves the robustness of OpenEdge Replication.

Before-image space management

Before-image (BI) files track OLTP changes critical for database functionality. In this release, enhancements let the OpenEdge database detect long-running idle transactions and prevent resource issues due to abnormal before-image file growth. This enhancement saves space and reduces performance issues due to bottlenecks in the BI file system. You can use the `-biscantime` startup parameter to control how long scans of BI clusters continue.

For more information see "Before-image scan time (-biscantime)" in *Manage the OpenEdge Database*.

Reorganize fragmented record data

The new PROUTIL TABLEREORG qualifier lets you reorganize fragmented record data while the associated table remains accessible to OLTP operations. The new process replaces lengthy dump and load operations, index rebuilds of associated indexes, and replication target rebase. Eligible record data must all be located within the same Type II storage area. Partitioned and multi-tenant tables are supported.

For details, see "PROUTIL TABLEREORG qualifier" in *Manage the OpenEdge Database*.

On-demand login broker changes

In this release, you can change the following startup parameters for secondary login brokers while the database is running:

- "Maximum Clients per Server (-Ma)"
- "Maximum Dynamic Server (-maxport)"
- "Minimum Clients per Server (-Mi)"
- "Minimum Dynamic Server (-minport)"
- "Message Buffer Size (-Mm)"
- "Maximum Servers per Broker (-Mpb)"
- "Pending Connection Time (-PendConnTime)"

See "Change database parameters while the database is running" in *Optimize the OpenEdge Database for Continuous Operation*.

Relaxed CRC rule for r-code

An application can now run existing r-code without recompiling if a new database field is added or modified provided the change does not change the OpenEdge data type of the number of extents being reference by the -r code.

For more information, see "Relaxed CRC validation rule for precompiled r-code" in *Use the Oracle DataServer*.

Simplified license changes

From this release onwards, changes to the OpenEdge control codes can be made quickly. You can also add control codes for 12.x report to recorded (R2R) bit products in an existing version of OpenEdge. These products include:

- OpenEdge CDC for Enterprise
- OpenEdge CDC for Workgroup
- OpenEdge Pro2 CDC for Workgroup
- OpenEdge Pro2 CDC for Enterprise
- OpenEdge Multi-Tenant Tables

- OpenEdge TDE
- OpenEdge Table Partitioning

For more information see, "Update existing product licenses" in *Install OpenEdge*.

Change ABL application schema without restart

Updates to deployed ABL applications with changes that depend on new schema elements do not require a restart of their ABL virtual machine (AVM) clients and agents to be able to execute the new code.

Developer experience

ABL enhancements

Here is a video that shows some of the enhancements made to ABL in OpenEdge 12.2 and 12.3:

https://www.youtube.com/embed/2M1Wb4BRh3Q

In OpenEdge 12.3, the following language enhancements are available.

New VAR statement

The VAR statement is a new ABL statement that allows developers to define program variables more concisely, in a manner similar to more modern programming languages. The VAR statement is a shorthand notation for the `DEFINE VARIABLE` statement. With VAR, you always get `NO-UNDO` behavior, multiple variables can be defined in a single statement, bracket notation is used for defining arrays, and the equal sign (=) is used to assign initial values.

```
DEFINE VARIABLE x AS INTEGER EXTENT 3
       NO-UNDO INITIAL [1, 2, 3].
```



```
VAR INTEGER[3] x = [1, 2, 3].
```

The following code shows some basic examples of the VAR statement:

```
/* Four character variables */
VAR CHAR s1, s2, s3, s4.

/* Date variable with an initial value */
VAR DATE d1 = 1/1/2020.

/* Object variable */
VAR mypackage.subdir.myclass myobj1.

/* Determinate array of size 3 */
VAR INT[3] a1 = [100, 200, 300].

/* Indeterminate array */
VAR INT[] a2.
```

For more information, see "Learn about the VAR statement" in *Develop ABL Applications* or "VAR statement" in the *ABL Reference*.

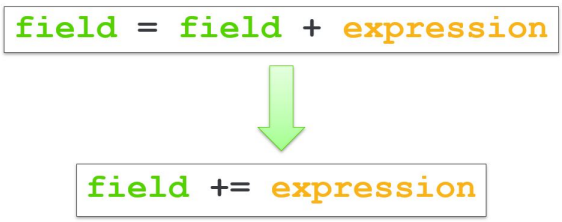
CHAR and INT keywords

The new VAR statement does not allow abbreviated keywords when specifying data types; data types must be fully spelled out. To allow for a more concise syntax with VAR, CHAR and INT are now considered full keywords. CHAR is a synonym for CHARACTER, and INT is a synonym for INTEGER. The new synonyms, CHAR and INT, can be used in all ABL statements. The use of CHAR and INT now satisfies the `require-full-keywords` compiler option.

For more information about the `require-full-keywords` compiler option, see the "COMPILE statement"in the *ABL Reference*.

New compound assignment operators

ABL introduces compound assignment operators (+=, -=, *=, /=) for performing an operation and assigning a value, using a shorthand notation. Supported operations include addition (numeric and date-based) and concatenation, subtraction (numeric and date-based), multiplication, and division.



For more information, see "Compound assignment operators" in *Basic ABL*.

Runtime resizing of indeterminate arrays

Previously, if an indeterminate array variable was defined with initial values and you tried to reset the size, a compiler error occurred. Now, you can resize indeterminate arrays at run time even if they are defined with initial values. The following example code illustrates the change:

```
// Define an indeterminate array with initial values
VAR CHARACTER[] myArray = ["a","b","c"].

MESSAGE "myArray has" EXTENT(myArray) "elements"
  VIEW-AS ALERT-BOX.  // myArray has 3 elements

EXTENT(myArray) = 10.  // can now reset the size (compiler error previously)

MESSAGE "myArray has" EXTENT(myArray) "elements"
  VIEW-AS ALERT-BOX.  // myArray has 10 elements
```

This feature applies to variable defined using both the VAR and `DEFINE VARIABLE` statements.

For more information, see "Work with one-dimensional arrays" in *Basic ABL*.

Register methods as event handlers for RUN calls

Developers can now register ABL object-oriented class methods as event handlers for ABL RUN asynchronous calls.

Before this release, only procedures could be used as event handlers for the ABL RUN ... ASYNC statement. With this feature, developers can use procedures or methods as event handlers. Public methods in classes are supported as callback handlers. As a part of this feature, the RUN statement is extended to support two new keywords, `EVENT-HANDLER` and `EVENT-HANDLER-CONTEXT`. These new keywords allow developers to specify either a procedure or a method as a callback handler.



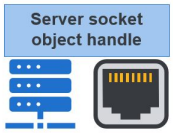
```
RUN ... ASYNCHRONOUS ...

  EVENT-HANDLER event-handler-name

  [ EVENT-HANDLER-CONTEXT event-handler-context ]
```

For more information, see the the following:

- "RUN statement" in the *ABL Reference*
- "Event procedures or methods" in the *ABL Reference*
- "EVENT-HANDLER attribute" in the *ABL Reference*
- "EVENT-HANDLER-OBJECT attribute" in the *ABL Reference*

Register methods as event handlers for server socket objects

Developers can now register ABL class methods as event handlers for the server socket `CONNECT` event. Before this release, only procedures could be used as event handlers for the server socket `CONNECT` event. With this feature, public methods are supported as callback handlers for server socket objects, allowing complete encapsulation of the server socket functionality into object-oriented ABL.




```
SET-CALLBACK (callback-name,
              routine-name [, routine-context])
```

For more information, see the following:

- "SET-CALLBACK( ) method" in the *ABL Reference*
- "CONNECT event" in the *ABL Reference*
- "Implementing an ABL socket server" in *OpenEdge Programming Interfaces*
- "Examples using ABL sockets" in *OpenEdge Programming Interfaces*

Enhanced CONNECTED () method for ABL clients

An enhanced server object handle `CONNECTED( )` method for ABL clients connecting to PAS for OpenEdge instances using the APSV transport is now available. Before OpenEdge 12.3, the method would return a logical value indicating whether the server handle was currently connected, but did not indicate whether there was a valid server session for the ABL client to connect to.

The server object handle `CONNECTED( )` client method was updated to determine if the server also supports the enhanced server object handle `CONNECTED( )` functionality. If both the client and the server support this functionality, then the ABL client sends an `HTTP HEAD` request to the server, and the request then:

- Confirms that there is a valid session for the client, and returns `true` or `false`.
- Resets the session inactivity timer, so that the request sent after the called `server-handle:CONNECTED( )` does not fail due to inactivity.

The following example code illustrates the updated behavior of the server object handle `CONNECTED()` method:

```
DEF VAR appHandle AS HANDLE.
DEF VAR ret       AS LOGICAL.
DEF VAR isConnected AS LOGICAL.

CREATE SERVER AppHandle.
ret = AppHandle:CONNECT("-URL http://localhost:8810/apsv -sessionModel Session-managed").

IF (ret) THEN DO:
 isConnected = AppHandle:CONNECTED().
 MESSAGE "connected: " isConnected VIEW-AS ALERT-BOX.
 IF (isConnected) THEN DO:
  // Server object has a valid session for ABL client.
 ELSE:
  // Server object does not have a valid session for ABL client.
END.
QUIT.
```

For more information about the `CONNECTED()` method, see "CONNECTED() method" in the *ABL Reference*.

Java and .NET Open Clients

New isConnected() method

OpenEdge 12.3 offers the new `isConnected()` method for instantiated `Progress.Open4GL.Proxy.OpenAppObject` instances for Java and .NET Open Clients, which is a logical value that indicates whether there is a valid session on the PAS for OpenEdge server for the requesting client.

The following example code illustrates the new `isConnected()` method in Java:

```
Connection connection = new Connection ("http://localhost:8810/apsv", "", "", "");
connection.setSessionModel( Connection.SM_SESSION_MANAGED );

OpenAppObject appobj = new OpenAppObject (connection, "");

if (appobj.isConnected()) {
     System.out.println("connected: true");
     // The session on the PAS for OpenEdge server is valid.
 // write code to run a procedure with this OpenAppObject.
} else {
     System.out.println("connected: false");
}
```

For more information, see "Return the validity of the server session on an OpenAppObject" in *Java Open Clients*.

The following example code illustrates the new `isConnected()` method in .NET:

```
Connection connection = new Connection ("http://localhost:8810/apsv", "", "", "");
connection.setSessionModel( Connection.SM_SESSION_MANAGED );

OpenAppObject appobj = new OpenAppObject (connection, "");

if (appobj.isConnected()) {
     Console.WriteLine("connected: true");
     // The session on the PAS for OpenEdge server is valid.
  // write code to run a procedure with this OpenAppObject.
} else {
```

```
    Console.WriteLine("connected: false");
}
```

For more information, see "Return the validity of the server session on an OpenAppObject" in *.NET Open Clients*.

SECPROP utility

In OpenEdge 12.3, the merge functionality in the SECPROP utility is enhanced with `-f` parameter.

For more information about the merge functionality, see 'SECPROP' in *Manage Progress Application Server (PAS) for OpenEdge*.

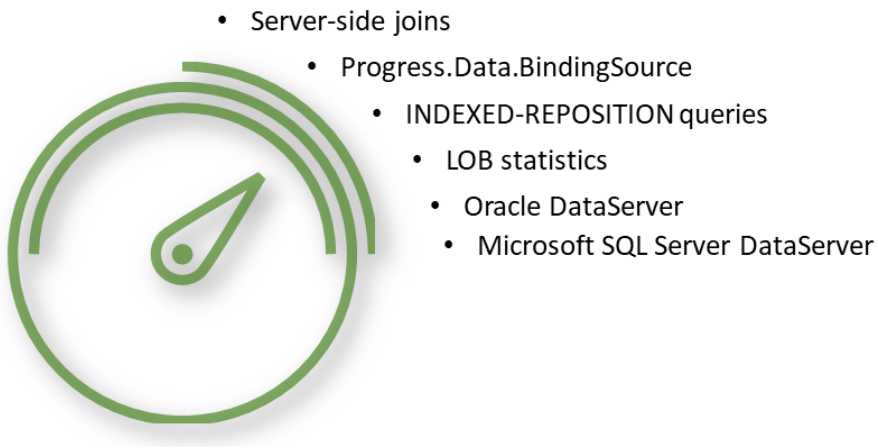
Third-party tools

In OpenEdge 12.3, the following third-party tools that are bundled with Progress Developer Studio for OpenEdge were updated. For more information about all third-party tools packaged with OpenEdge, see the OpenEdge Product Availability Guide on the *Progress Information Hub*.

Infragistics controls update

The "OE Ultra Controls .NET" license ships "Infragistics Ultimate UI for Windows Forms". For OpenEdge 12.3, the version was updated to 19.2.20192.381.

Performance



Enhanced server-side join processing

Server-side join improves performance by reducing the amount of data transferred across the network and by reducing the computational cost on the client. Server-side join is available in a multi-threaded database server, and is enabled for queries where the tables to be joined are in the same logical database and there are 10 or fewer tables. Server-side join only occurs when using the following ABL constructs:

- `FOR` statements with `NO-LOCK`

- Static or dynamic queries with NO-LOCK, except for those that use the INDEXED-REPOSITION option or involve an outer join

Although the query needs to be NO-LOCK for the server-side join to be in effect, you can obtain records with a lock using the GET statement for a static query, or using the GET method (for example, GET-NEXT( )) for a dynamic query.

For more information, see "Server-side join processing" in *OpenEdge Programming Interfaces*.

Improved database query operations

The new SyncToSelectedRow property can be used to control whether the query cursor is repositioned to the selected row after fetching data from another row. The default value is true. By setting the value to false, you can speed up operations that do not require repositioning, for certain use cases.

For more information, see "SyncToSelectedRow property" in the *ABL Reference*.

Enhanced index repositioning

Optimized SQL can now be generated for INDEXED-REPOSITION queries when the index sort order and the result sort order is opposite.

For more information, see "Index repositioning" in *Use the Oracle Data Server*.

PROUTIL enhancement for LOB statistics

The new PROUTIL LOBANALYS qualifier returns information for LOB only. Previous OpenEdge releases included LOB information with table record statistics reports. PROUTIL LOBANALYS collects LOB statistics by table or area, where the area can be the specific LOB area for a table, and skips the collection of table or database-wide statistics.

For details, see "PROUTIL LOBANALYS qualifier" in *Manage the OpenEdge Database*.

Savepoint support

Savepoint support was implemented in the Oracle DataServers. The sub-transactions containing changes to BLOB/CLOB fields can now be undone without undoing the entire transaction. This improves user experience with Oracle DataServers.

For more information, see "Handling rollback of sub-transactions" in *Use the Oracle DataServer*.

FOR FIRST support

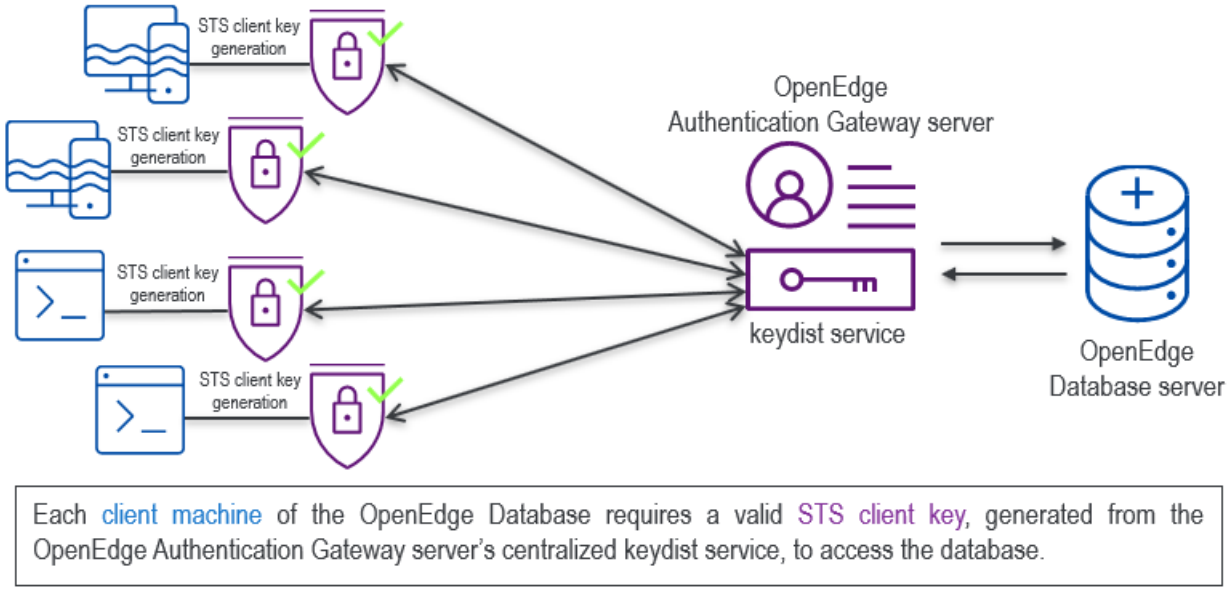
When coding ABL applications for the Microsoft SQL Server DataServer, FOR buffer statements can now be replaced with FOR statements when retrieving a single record.

For more information, see "Replacing FOR FIRST for performance" in *Use Microsoft SQL Server DataServer*.

Security

Remote STS client key management

OpenEdge 12.3 includes a key distribution application that provides remote STS client key management for OpenEdge Authentication Gateway installations. If you configure your Authentication Gateway server to use an STS server key, each remote OpenEdge installation must generate a corresponding STS client key. The STS client key generation must happen on the client system, and requires the client system to have access to the STS server key. With earlier versions of the Authentication Gateway, this was done though a file share on a remote drive, or by manually copying the STS server key to the client machine. This was a manual process that could leave the STS server key on disk.



OpenEdge 12.3 offers an STS client key management service that runs on the Authentication Gateway through the key distribution application. When you install this service on the Authentication Gateway server using the key distribution application (`keydist.war`), STS client key generation can be done on the client machine by specifying the URL of the Authentication Gateway server and providing the proper credentials. This can be done through the stskeyutil tool or through the STSKey plugin for the OpenEdge AdminServer.

For more information, see the following topics in the *OpenEdge Authentication Gateway Guide*:

- "About STS client key management"
- "Configure STS client key management using stskeyutil"
- "Configure STS client key management using the AdminServer STSKey Plugin"
- "STS client key management logging"

Installation support for client key management

The OpenEdge installation program optionally allows the specification of an OpenEdge Authentication Gateway server and password that will be used to generate a valid client key on the system being provisioned. This option is available for any products that support authentication through the Authentication Gateway.

For more information, see "Select your OpenEdge installation choices" in *Install OpenEdge*.

OpenEdge security domain concepts

OpenEdge security domains are modeled on a security domain pattern where servers, applications, and databases all trust a common security token that is issued to an authenticated user account that is recognized as a valid user of the domain. Having a fundamental understanding of what OpenEdge security domains are, how they work, and how to configure them, is essential for using them effectively within your OpenEdge application, especially in production environments.

A new document is available on the Progress Information Hub in this release of OpenEdge that explains essential concepts, and also includes a description of how they work in real time to control access.

For more information, see "Introduction to OpenEdge security domains" in *Learn About OpenEdge Security Domains*.

OpenEdge SQL

Protrace extension on Windows

Currently, the `protrace.`*`pid`* file only contains a stack trace of the SQL process that terminated abnormally on some platforms, making it difficult to quickly identify the specific SQL statement responsible for the crash.

As with all other platforms, a Protrace file will capture diagnostic information such as a SQL query that results in a crash apart from a stack trace, even on Windows.

Note:

In 12.3, you cannot turn this feature ON using a start-up parameter, instead you use an “Enhanced SQL Logging statement”.

To capture details of queries that are generating errors, enable “SQL logging”. This captures information about all the queries that were run. This results in a large SQL log and makes it difficult to browse through. With the capability of “SQL failure-driven logging”, only the failed query information is logged proactively.

For more information, see "Troubleshoot an SQL Crash" in *OpenEdge Data Management: SQL Development*.

Generate core dumps for SQL servers

The `-SQLEnableCoreGeneration` parameter can be used to generate a core dump when the OpenEdge SQL server crashes.

Note: This is applicable only on the Linux platform.

For more information, see "Database server performance parameters" in *Manage the OpenEdge Database*.

Pro2 6.2 New Features

Database migration

To stay competitive, your applications need to run as close to 24/7 as possible. In some cases, you can only have downtime limited to 4 hours. This can make upgrading or migrating your OpenEdge database difficult, even with proper planning. With Pro2 version 6.2, you can migrate your OpenEdge database with minimal downtime. In this video, we show you how you can migrate with Pro2.

https://www.youtube.com/embed/z6Oluo2enN4

CDC enhancements

In previous versions of Pro2, the CDC replication process converted CDC records to Pro2 `replqueue` records. This procedure increased the total time of the replication.

In Pro2 6.2, CDC replication records are consumed directly by Pro2, bypassing the need for the records to be converted to `replqueue` records.

The direct replication of CDC works on all Pro2 target database types: Microsoft SQL Server, Oracle Database, and OpenEdge. This enhancement also functions on both network topologies, LAN and WAN.

For more information, see Use CDC with Pro2.

Bulk-load enhancements

In previous releases of Pro2, bulk-load process used row-by-row transactional commits on target replication database.

In Pro2 6.2, the bulk-load process is optimized to directly generate native SQL INSERT statements and wrap multiple inserts into a single target transaction. This enhancement has improved the efficiency of the loading process. For table loads that run for days or weeks, this overall improvement can be dramatic.

Because the new bulk-load method is a Pro2-produced methodology, this new feature operates identically in both Oracle Database and Microsoft SQL Server environments, allowing Pro2 to provide wide transaction support in Microsoft SQL Server environments.

For more information, see Bulk load with Pro2.

Release notes

Issues fixed in OpenEdge 12.3

Click here to see a list of issues fixed in OpenEdge 12.3.

Known issues in OpenEdge 12.3

Click here to see a list of known issues in OpenEdge 12.3.

4

What's New in OpenEdge 12.2

OpenEdge Release 12.2 focuses on the following themes:

- Continuous operations on page 38
- Performance improvements on page 41
- Security updates on page 42
- Developer experience enhancements on page 43

This content provides updates about the following topics:

- Java support on page 49
- OpenSSL, TLS, and Cipher Suite support on page 43
- OpenEdge Adapters on page 51

This content also describes enhancements to the following OpenEdge components:

- PAS for OpenEdge on page 49
- OpenEdge Database on page 49
- OpenEdge Management on page 52
- OpenEdge SQL on page 52
- OpenEdge Pro2
- De-supported and deprecated features in OpenEdge 12.2 on page 54

Important: New Relic plugins for PAS for OpenEdge and the OpenEdge Database.

As of June 2021, OpenEdge discontinued support for the NewRelic plugins for PAS for OpenEdge and the OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to the view plugin dashboard. This limitation is a result of the dropped support by New Relic of the OpenEdge plugins as of Wednesday, June 16th, 2021. The official announcement is available from https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

For details, see the following topics:

- Continuous operations
- Performance improvements
- Security updates
- Developer experience enhancements
- Java support
- OpenEdge Database
- PAS for OpenEdge
- OpenEdge Adapters
- OpenEdge Management
- OpenEdge SQL
- OpenEdge Pro2 6.1
- De-supported and deprecated features in OpenEdge 12.2

Continuous operations

OpenEdge 12.2 delivers a range of enhancements that support continuous operation.

Automatic database reconnection

OpenEdge now provides a mechanism for automatically connecting to an alternate database, if the connection to the primary database fails. This mechanism uses new client connection startup parameters (`-dbalt1`, `-dbalt2`, `-retryConnect`, `-retryConnectPause`) and is available when using a client-server connection.

For more information, see "Learn about automatic database reconnect" in *Manage ABL Applications*.

Reconnect on application restart

`-autoReconnect` is a new client session startup parameter related to the automatic database reconnect feature; however, it only applies to non-batch mode AVMs. When a database disconnect occurs, the AVM raises a STOP condition. If the application does not handle the STOP condition on its own, then for an interactive client, the AVM restarts the application. However, the AVM does not attempt to re-establish database connections that were initially connected using the startup parameters. `-autoReconnect` changes this behavior and instructs the AVM to attempt to re-establish lost database connections before restarting the application.

For more information, see "Learn about automatic database reconnect" in *Manage ABL Applications*.

Brokers run throughout recovery transitions

To finish transitions faster, database brokers now continue running on single targets and replication sets. This change improves both manual and automatic recovery transitions.

For more information, see "Plan for transition" in *Use Database Replication*.

Disconnect secondary brokers from PROMON

A new PROMON option lets you disconnect secondary brokers without issuing a process kill signal. Secondary brokers let you manage the configuration for a specific group of end users, such as ABL clients, SQL clients, or both.

This option reduces potential database downtime.

For more information, see "Server Options" in *Manage the OpenEdge Database*.

Control database notification operations

The new `-dbnotifyops` startup parameter supports concurrent schema updates. The parameter controls the maximum number of concurrent notifications pending.

For more information, see "Database Notification Operations (-dbnotifyops)" in *Manage the OpenEdge Database*.

Manage database client notification

To prevent unnecessary network traffic and improve response time, notifications of certain schema changes are automatically sent to remote clients. Startup parameters, PROMON utility, and VST changes let you control the method and frequency of notification delivery.

For more information, see "Database Notification Time (-dbnotifytime)" in *Manage the OpenEdge Database*.

Change more database parameters while the database runs

In this release, more parameters can be changed while the database is running. Parameters that now allow such changes are `-aiarcdir`, `-dbnotifyops`,`-dbnotifytime`, `-indexrangesize`, `-lobrangesize`, `-Ma`, `-Mm`, `-Mn`, `-n`, `-semsets`, `-ssj`, `-tablerangesize`, `-userindexrangesize`, `-userlobrangesize`, and `-usertablerangesize`.

For a list of all parameters that you can modify online, see "Change database parameters while the database is running" in *Optimize the OpenEdge Database for Continuous Operation*.

Change ABL triggers online

You can create, change, or delete ABL triggers without acquiring an exclusive schema lock.

For more information, see "Change ABL triggers online" in *Optimize the OpenEdge Database for Continuous Operation*.

PROMON server screen changes support separate server and user numbers

To support continuous operation, the server number and the server's user number are now stored as separate fields: `Sv No` and `SvUsr No`. These fields, which apply only to remote clients, display on the:

- **User Control** screen
- **Servers** and **Servers by broker** screens in R&D Status Displays
- **Terminate a Server/Secondary Broker** screen in R&D Administrative Functions

VST changes support separate server and user numbers

To support continuous operation, the server number and the server's user number are now stored as separate columns in virtual system tables (VSTs) as follows:

Table name	Changed column	Purpose
_Connect	_Connect-Server	Server number
	_Connect-ServerUsrNum	Server's user number
_Server	_Server-Num	Server number
	_Server-UsrNum	Server's user number

The changed columns apply only to remote clients.

Database log file specifies both server and user numbers

The OpenEdge database log (`.lg`) now shows the server number and user number for each connected remote server or broker.

For an example of multiple, concurrent server connections, see Find connected brokers and servers in *Manage the OpenEdge Database*.

Database log file records secondary broker parameters

The database log (`.lg`) captures information that helps you understand the circumstances surrounding a failure. The log now records startup parameters for secondary brokers.

Although you can list broker parameters for running servers by using PROMON R&D Status or Virtual System Tables, the database log file lets you troubleshoot landscape or connection issues after server failure. You can easily determine which brokers were active during an event.

For more information about the database log, see "OpenEdge database log file" in *Manage the OpenEdge Database.*.

PAS for OpenEdge continuous operations updates

Progress Application Server (PAS) for OpenEdge offers a new packaging and deployment strategy that supports the use of ABL applications in continuous integrations and continuous delivery (CI/CD) pipelines called OpenEdge Application Archives (OEAR).

Rename fields using SQL

Renaming a database field using SQL no longer requires an exclusive schema lock or for users to be disconnected from the database.

For more information, see "Rename fields using SQL" in *OpenEdge Data Management: SQL Development*.

Rename fields using ABL

Renaming a database field using ABL no longer requires an exclusive schema lock or for users to be disconnected from the database.

For more information, see "Rename fields using ABL" in *Optimize the OpenEdge Database for Continuous Operation*.

Performance improvements

OpenEdge 12.2 delivers new performance enhancements, which are described in this topic.

Expansion of server-side join processing includes non-FORWARD-ONLY dynamic queries with NO-LOCK

Server-side join improves performance by reducing the amount of data transferred across the network and by reducing the computational cost on the client. Server-side join is available in a multi-threaded database server, and is enabled for queries where the tables to be joined are in the same logical database and there are 10 or fewer tables. In addition, server-side join only occurs when using the following ABL constructs:

- FOR statements with NO-LOCK.
- Dynamic queries with NO-LOCK, except for those that use the INDEXED-REPOSITION option or involve an outer join.

For more information about server-side join, see "Server-side join processing" in *OpenEdge Programming Interfaces*.

Reusable objects cache now includes objects containing datasets and events

When an object-oriented ABL object is destroyed (whether explicitly deleted or through garbage collection), it is generally placed in the reusable objects cache. Reusing objects from this cache saves the overhead of instantiating a new class. Before this release, classes containing a dataset or event definition (DEFINE DATASET or DEFINE EVENT) were disqualified from use in the cache. Now, with this release, these objects are included in the cache. As a result, users may see a improved performance.

For more information about the reusable objects cache, see "Re-usable Objects Cache (-reusableObjects)" in *Startup Command and Parameter Reference*.

Enhanced performance of ABL profiler

Starting with this release, the ABL virtual machine (AVM) conducts in-time analysis and no longer writes the timing data in a temporary file. This results in performance gains because it bypasses the input-output operations on the disk. It also results in optimized disk utilization, especially with long running applications and queries, because temporary files can be lengthy.

New ABL Profiler content

Profiling offers ABL developers the ability to collect meaningful data about their program execution. The information collected identifies potential performance bottlenecks and gives developers more visibility into which parts of their applications are taking longer to execute. In prior releases, the "Profiler View" in *Progress Developer Studio for OpenEdge Online Help* and how to perform "server-side profiling" in *Manage Progress Application Server (PAS) for OpenEdge* were documented. In this release, new content is available for the following:

- "Identify ABL performance problems with profiler" in *OpenEdge Videos*
- "PROFILER system handle" in *ABL Reference*
- "-profile startup parameter" in *Startup Command and Parameter Reference*

Enhanced ABL session management in PAS for OpenEdge

PAS for OpenEdge has enhanced the multi-session agent so that it self-manages ABL sessions. This means that requests no longer fail due to resource unavailability.

For more information, see "Enhanced ABL session management" in *Manage PAS for OpenEdge*.

Enhanced INDEX function to allow search of a string stored in a MEMPTR variable

The INDEX function returns an INTEGER value that indicates the position of the searchString within the source string. The source string now includes a MEMPTR expression along with CHARACTER and LONGCHAR variables.

For more information, see "INDEX function" in *ABL Reference*.

Security updates

To provide continuous improvements to its security features support, OpenEdge 12.2 adds the following features:

Safe User ID

This release introduces the Safe User ID feature, which is provided for ABL applications that access multi-tenant databases through Progress Application Server (PAS) for OpenEdge application servers. The Safe User ID feature helps maintain the synchronization between the database's current user identity and the client for whom the tenant data is being accessed, thereby preventing the tenant's data from becoming available to a user of a different tenant.

A safe user is an identity that has no tenant data access but has the required access privileges to shared data. Ideally, the safe user should be a placeholder identity that is defined in the user account store that is used by the PAS for OpenEdge instance in which the ABL application executes.

This feature provides a safeguard against exposing incorrect tenant data when processing a client request. It enables a mechanism that resets the database connection to this safe user identity when processing each request on the application server. This forces the application to set the connection to a valid tenant user identity for a client request that requires access to tenant data in the database.

To support this feature, OpenEdge provides the following new functionality:

- The ability to enable the Safe User ID feature on an OpenEdge database instance that is configured for multi-tenancy.

  When the database administrator enables this feature, the administrator selects a policy that determines *when* the safe user identity is set on the connection to the database. The policy corresponds to the execution of either the Activation or Deactivation event policies in the ABL application, which are invoked in response to a client request.

- Two new methods for the "SECURITY-POLICY" system handle are described in the *ABL Reference*:
  - GET-SAFE-USER( )—Returns the handle to a copy of the sealed client-principal object that represents the safe user identity for the specified multi-tenant database connection.
  - SET-SAFE-USER( )—Sets a multi-tenant database connection's current user identity to the stored safe user identity for the database.

For complete information, see "Overview of Safe User ID" in *Learn about Identity Management*.

TDE support for new passphrase recoverable Database Master Key algorithms

At Transparent Data Encryption (TDE) enablement, you can now choose either of two new passphrase recoverable Database Master Key (DMK) key derivation algorithms that offer different key sizes for data security:

- Advanced Encryption Standard (AES) 128-bit DMK key generation
- Advanced Encryption Standard (AES) 256-bit DMK key generation

The previous algorithm choice of a passphrase recoverable DMK is deprecated, but you can still use it during a TDE keystore recovery operation. The new selections are not backward compatible.

The TDE keystore recovery process still requires that you use exactly the same recoverable passphrase and DMK passphrase key algorithm, whether you choose the original or one of the newer algorithm versions.

For more information, see "Execute the ENABLE ENCRYPTION command" in *Learn About Security and Auditing*.

Improved hashing algorithm for TDE keystore passphrases

To increase passphrase protection, OpenEdge databases use an enhanced hashing algorithm for Transparent Data Encryption (TDE) key store passphrases. In OpenEdge 12.2, newly enabled databases automatically use the new algorithm, and updates to the user or admin passphrase automatically change the algorithm to the enhanced version.

For more information, see "Modify passphrases" in *Learn About Security and Auditing*.

Relaxed length restriction for DataServer

Starting with this release, DataServer relaxed the length restrictions for usernames and passwords for Microsoft SQL Server and Oracle DataServer to improve security. For Microsoft SQL Server, the allowed length for user ID and password is 32 and 128 characters, respectively. For Oracle, the allowed length for user ID and password is 30 characters each.

For more information, see "Authorization and authentication" in *OpenEdge Data Management: DataServer for Microsoft SQL Server* and "Authorization and Authentication for connecting a schema holder" in *Use the Oracle Data Server*.

OpenSSL, TLS, and Cipher Suite support

For more information, see OpenSSL, TLS, and Cipher Suite support.

OpenSSL, TLS, and Cipher Suite support

OpenEdge 12.2 supports SSLv3, and TLS versions 1.1, 1.2, and 1.3. The default is TLS 1.2.

In OpenEdge 12.2, the transport layer security was strengthened by upgrading OpenSSL to 1.1.1d.

For more information, see Cipher Suites for OpenSSL, TLS support in Java for OpenEdge, and TLS support in .NET for OpenEdge.

Developer experience enhancements

Progress continues being committed to improving the developer experience through tooling and language enhancements.

ABL enhancements

The ABL continuously evolves to support customer-requested features. In OpenEdge 12.2, the following language enhancements are available.

- **Package-level protection** — New package-level access modes, `PACKAGE-PRIVATE` and `PACKAGE-PROTECTED`, are available. Access modes provide the ability to hide or expose class members to other parts of an application based on how the class member is defined. The new modes provide the ability to extend access outside of the class hierarchy to classes within the same package. A package is

the directory structure, relative to the PROPATH leading up to the class itself, as specified in the CLASS statement. This directory path identifies the location of the file that defines a class, interface, or enumeration. PACKAGE-PRIVATE and PACKAGE-PROTECTED restrict access at the package level which is helpful for developers designing frameworks and reusable libraries.

For more information, see the "ABL Syntax Reference" and "Access modes" in *Develop Object-oriented ABL Applications* for updates and default values for applicable class members. Progress Developer Studio for OpenEdge exposes these new access modes when defining and displaying class members, where applicable.

- **Require-return-values compiler option** — A new compiler option is available, that helps ensure all logical code paths have RETURN *value* statements in user-defined functions, non-VOID methods, and property getters. Missing RETURN *value* statements may lead to incorrect behavior and code that is hard to debug. Using the option helps developers identify and fix problems, resulting in more robust code.

  For more information, see "Require-return-values compiler option" in *Basic ABL*.

- **Ability to use methods as callbacks for socket objects** — Developers can now associate a method within a class with an ABL callback event for handle-based socket objects. This functionality is already part of other language elements such as ProDataSets, Buffers, and Queries. Before this release, only procedures could be used for socket object callback events. With this feature, developers can use procedures and both regular and static methods as the callback handler for a socket, allowing complete encapsulation of the client socket functionality into object-oriented ABL.

  For more information, see the "SET-CALLBACK() method" in the *ABL Reference* and "Socket example using a method as a callback" in *OpenEdge Programming Interfaces*.

Java and .NET Open Clients

Inner exception errors are now maintained when errors occur using the Java and .NET Open Client library. The goal of this change is to make error handling more accurate and to improve developer debugging experience.

Host Language Call Interface

The Host Language Call (HLC) interface is now supported both by the single-threaded ABL client and the PAS for OpenEdge multi-session agent using a DLL (on Windows) or a shared object (on UNIX). Consequently, if you want to deploy an ABL procedure that contains the CALL statement to a PAS for OpenEdge instance, you must update the hlprodsp.c file to indicate whether or not your C source code is thread safe. You cannot build custom ABL client executables by statically linking C code.

For more information about the enhanced HLC functionality, see "HLC and OpenEdge" in *OpenEdge Programming Interfaces*.

Progress Developer Studio

Progress Developer Studio for OpenEdge Release 12.2 takes advantage of Eclipse version 4.13, and incorporates many of the feature enhancements included in this version of Eclipse.

For more information about Eclipse feature enhancements for version 4.13, see the Eclipse website.

The following list is a snapshot of the features included in Progress Developer Studio for OpenEdge Release 12.2:

- Content assist for the workspace—This feature is applicable only in the workspace selection window. Press CTRL+Space to display the list of workspaces.
- New project creation link—The **Create a new OpenEdge project** link is added in the **Project Explorer** panel. Click the link to open the **New OpenEdge Project** wizard. This link is visible only if there are no projects added in the **Project Explorer**.

- New **User filters** tab—This tab is added to the **Filters and Customization** section. With this feature, you can create an added filter to hide all the files that have a specific extension in the **Project Explorer** view. To do this, add the extension in the **User filters** tab and click the checkbox to activate it.
- New shortcut for **Quick Search**—Press ALT+CTRL+SHFT+L from anywhere in a workspace to launch the **Quick Search** wizard.
- Ability to save files in the editor when the focus is on another panel—Save an open file by clicking **Save** from the **File** menu or by pressing the CTRL+S, even if the focus is on another panel and not on the editor.
- New **Show In** option in the **Project Explorer**—Right-click a project or a file, and select **Show In**. Select **Properties** to open the **Properties** tab. Click the **System Explorer**, and the project or file location opens in Windows Explorer.
- New **Show Command Line** button in the **External Tools Configurations** wizard—Click the **Show Command Line** button to view how the command is passed in the command line.

UBPF configuration file

The **Unified Build and Packaging Framework (UBPF)** is a Gradle plugin that allows you to seamlessly transition between the development and integration environments when you build your ABL project outside of Progress Developer Studio for OpenEdge. This framework enables you to have the same environment setup for both your development and integration environments.

To work with the **UBPF** plugin, you use a configuration file, `ubpf.config`, that is generated when you create an ABL project in Progress Developer Studio.

Note: Unified Build and Packaging Framework (UBPF) is renamed to the OpenEdge DevOps Framework (OEDF).

For more information about the configuration file, see the "Unified Build and Packaging Framework" in *Progress Developer Studio for OpenEdge Online Help*.

Bundled third-party tool updates

In OpenEdge 12.2, the following third-party tools that are bundled with Progress Developer Studio for OpenEdge were updated. For more information about all third-party tools packaged with OpenEdge, see the OpenEdge Product Availability Guide on the *Progress Information Hub*.

- **Infragistics controls update**—The "OE Ultra Controls .NET" license ships "Infragistics Ultimate UI for Windows Forms". For OpenEdge 12.2, the version was updated to 19.2.20192.198.
- **Code Analyzer for ABL**—The Code Analyzer for ABL (CABL) plugin is shipped with OpenEdge 12.2. The version is updated to 4.5.
- **Progress Compilation Tool**—The Progress Compilation Tool (PCT) plugin is also shipped with OpenEdge 12.2. The updated version is 215.

Updates to ABLUnit

Application code that extends or calls any classes in the `OpenEdge.ABLUnit` namespaces may need to be recompiled or updated to use the new public API (classes and methods). The following changes require the modification and removal of a number of public members:

- Methods with lifecycle annotations
- Lifecycle method failure and error reporting
- `errNum` key for expected error numbers
- Failure and errors written to `results.xml`
- `<skipped>` element for ignored tests
- Stop conditions caught
- `<properties>` element for test suites
- `@Testsuite` or `@Test` annotations required

Note: All output produced in the `results.xml` file is consistent with the JUnit results standard.

Methods with lifecycle annotations

The ABLUnit test framework has the concept of lifecycle methods that add behavior to test cases and individual tests, for instance, code that must run before each test method. Previous releases allowed multiple methods with the same annotation to be run, but at run time, only a single lifecycle method was run.

Now, all of the methods that have lifecycle annotations are run in the order in which they are defined. Methods must all have the same signature (return void, no parameters), but the `OOABL` class requires the methods to have different names.

In the following example, `Setup_A` and `Setup_B` both run before `Test1` runs.

```
class Tests.MyTest:
    @Test.
    method public void Test1():
    // test method
    end method.

    @Setup.
    method public void Setup_A():
        // setup logic
    end method.

    @Setup.
    method public void Setup_B():
        // setup logic
 end method.
end class.
```

In order for multiple methods with lifecycle annotations to be run, additional behavior changes were made.

Lifecycle method failure and error reporting

A test that returns `OpenEdge.Core.AssertionFailedError` is a failure because a particular assertion was tested. All other conditions raised are considered errors.

If a test method has one or more lifecycle methods, and one of those methods (either the test method itself or one or more of the lifecycle methods) raises any error, then the most severe condition raised is reported. For example, if the test method raises a failure due to an assertion failing, and the lifecycle method raises an error, then the error is reported because an error is more severe than a failure.

errNum key for expected error numbers

The `@Test` annotation enables developers to specify the type of exception raised using an expected key with an OOABL type name value. To specify an error number, use the `errNum` key. The value for `errNum` must be an integer. The test passes if the following are true:

- The test raises an exception, and the exception matches the expected type.
- An `errNum` value is provided.
- The exception's first error number equals the `errNum` value.

If no `errNum` key is provided, then only the expected type is used to determine a test's success.

For example, the following test passes because it does not expect the class to exist.

```
@Test(expected="Progress.Lang.SysError", errNum=15287).
method public void Test_ClassNotFound():
    define variable obj as Progress.Lang.Object no-undo.
    obj = dynamic-new 'This.Class.Does.Not.Exist' ().
end method.
```

Failures and errors written to results.xml

A single test may have zero, one, or more lifecycle methods associated with it. The `@Setup` and `@TearDown` methods run before and after each test method, respectively. Failures or errors resulting from the test method or its lifecycle methods each have a failure or error element created in the `results.xml`.

For example, there are two failures in the test named `test3`:

```
<testcase classname="TestRunFailure.AfterEachFailure" name="test3"
          status="Error" time="0">
    <failure message="Expected: TRUE but was: no"
             type="OpenEdge.Core.AssertionFailedError">
        IsTrue OpenEdge.Core.Assert at line 177
    </failure>
    <error message="** Invalid character in numeric input T. (76)"
           type="Progress.Lang.SysError">
        TestRunFailure.AfterEachFailure at line 9
    </error>
</testcase>
```

<skipped> element for ignored tests

If a test is ignored, then a `<skipped>` element is added with a `message` attribute.

```
<testcase classname="TestRunFailure.BeforeAllFailure" name="test"
          status="Success" time="0">
    <skipped message="Ignored" />
</testcase>
```

Stop conditions caught

ABLUnit handles stop conditions raised in test methods (explicitly or implicitly). The message written to the `results.xml` file varies depending on the value of the `-catchStop` parameter.

If the `-catchStop` parameter has a value of 0, then the `Progress.Lang.AppError` error is raised and caught, and the message "stop condition raised for *`test-name`*" is added as an error to the test result.

If the `-catchStop` parameter has a value of 1, then the `Progress.Lang.Stop` exception is caught, and an error message is added to the test results. The message is typically "Stop condition raised" except for lock conflicts. If there is a lock conflict, then the message is "Lock conflict raised".

<properties> element for test suites

A `<properties>` element is added to `<testcase>` (test classes and procedures) and `<testsuite>` elements. This element contains a number of properties that describe the session.

The following is a list of properties and the ABL functions or attributes used to determine their values:

```
<properties>
    <property name="proversion"           value=proversion(1) />
    <property name="process-architecture" value=process-architecture />
    <property name="opsys"                value=opsys />
    <property name="window-system"        value=session:window-system />
    <property name="cpinternal"           value=session:cpinternal />
    <property name="startup-parameters"   value=session:startup-parameters />
</properties>
```

@TestSuite or @Test annotation required

Only programs (classes or procedures) that have an `@TestSuite` annotation or at least one `@Test` annotation are considered tests, and are added to the list of tests to run. All other programs are ignored by the ABLUnit test runner.

For more information about the ABLUnit framework, see "Lifecycle of ABLUnit framework" in *Progress Developer Studio for OpenEdge Online Help*.

Support for deploying an OpenEdge database in a Docker container

To enable Progress partners and users to more efficiently implement their continuous integrations and continuous delivery (CI/CD) pipelines, there is now support for deploying an OpenEdge database in a Docker container. For example, you can create and start a database in a Docker container on demand, run your tests, and then dispose of the container immediately after your tests are executed. As a result, you can focus on optimizing continuous testing of code increments without worrying about deployment details such as system resources, availability of a dedicated machine, and database setup.

For more information, see *Learn About OpenEdge Database Docker Container Images*.

Note: The OpenEdge database Docker container images are to be used in a development environment only. **DO NOT** use the images in a production environment.

Java support

OpenEdge 12.2 supports Java 11, a long-term support release of Java. The benefit to you is the ability to receive security updates and extensions to functionality. When Java updates become available, you decide whether or not to update your OpenEdge installation. This means that your OpenEdge applications and third-party JAR files have the following Java support commitments:

- Standard support until September 2023
- Extended support until September 2026
- Sustaining support indefinitely

Frequently asked questions

An FAQ topic is available to track a list of Java 11 related questions.

For more information, see "Java 11 support for OpenEdge FAQ" in *OpenEdge Frequently Asked Questions*.

OpenEdge Database

Progress continues being committed to improving the database administrator's (DBA) experience through OpenEdge database enhancements.

Help for database utilities

OpenEdge database utilities have subcommands and options that result in hundreds of syntax combinations. Misspelling or omitting a parameter may cause unexpected results or frustrating delays in order to look up the correct syntax.

Several OpenEdge database utilities now provide a help option that displays utility syntax at the command prompt. The `-help` option allows you to check syntax for the subcommands and subutilities at the command line, instead of stopping to look up the syntax on the content portal or in printed documentation. You can even save the command output and search on it.

For more information, see "Show syntax help for database utilities" in *Manage the OpenEdge Database*.

Advanced options in the PROMON utility

The PROMON utility offers a new screen for developers and database administrators. The submenus of **Advanced R&D** list database information that is helpful for monitoring and analyzing performance.

For more information, see "PROMON R&D Advanced Options" in *Manage the OpenEdge Database*.

In addition to these features, the OpenEdge database continues to enhance support for continuous operation.

For more information, see Continuous operations in this document.

PAS for OpenEdge

Progress Application Server (PAS) for OpenEdge offers the following new features in OpenEdge 12.2:

Create and deploy OpenEdge Application Archive (OEAR) packages

The following new topics were written in support of the OpenEdge Application Archive (OEAR) packaging and deployment structure:

- **OpenEdge Application Archive Structure**

  Details about the requirements for the structure of OEAR packages are explained in the "OpenEdge Application Archive Structure" topic in *Manage PAS for OpenEdge*.

- **Create an OpenEdge Application Archive using TCMAN export**

  The OpenEdge Application Archive (OEAR) is a new application packaging strategy where an ABL application and its components are packaged as a unit of deployment. This packaging simplifies ABL application deployment, and supports ABL application deployment in CI/CD pipelines.

  For more information, see "Create an OpenEdge Application Archive using tcman export" in *Manage PAS for OpenEdge*.

- **Create an OpenEdge Application Archive using an Ant Build**

  PAS for OpenEdge now includes an OpenEdge Ant Task called ABLApp that allows system administrators or DevOps engineers to automate the creation of OpenEdge Application Archive (OEAR) packages for use in CI/CD pipelines.

  For more information, see "Create an OpenEdge Application Archive using an Ant Build" and "Tailor an ABLApp installation using Ant" in *Manage PAS for OpenEdge*.

- **Deploy an ABL application as an OpenEdge Application Archive**

  PAS for OpenEdge has introduced a new application deployment strategy where an ABL application and its components are packaged as an OpenEdge Application Archive (OEAR). This new deployment strategy supports ABL application deployment in CI/CD pipelines.

  For more information, see "Deploy an OpenEdge Application Archive using tcman import" in *Manage PAS for OpenEdge*.

Enhanced ABL session management

PAS for OpenEdge enhanced the multi-session agent so that it self-manages ABL sessions. This means that requests no longer fail due to resource unavailability.

For more information, see "Enhanced ABL session management" in *Manage PAS for OpenEdge*.

Deploy new web handlers in OpenEdge 12.2

Web handler deployment changed in OpenEdge 12.2. The existing deployment model for web handlers is still supported, but a new web handler model was added that can help mange large numbers of APIs and does not require edits to the `openedge.properties` file.

Using the new functionality, web handlers are no longer specified in `openedge.properties`, but through an information file for each web service called either `ROOT.handlers` or *`webapp-name`*`.handlers`.

For more information, see "Deploy web handler services" in *Manage PAS for OpenEdge*.

PAS for OpenEdge new production security model

PAS for OpenEdge added a third security model to the product, which is a subset of the Production security model. The security model is called `pas` and it is accessed at instance creation by using the `-Z pas` option. Specifying the `pas` production security model creates an instance with a WAR file called `noaccess.war` that is deployed to the ROOT application, instead of `oeabl.war`. Additionally, instances with this security model have no default ABL application deployed, and therefore does not have a multi-session agent.

For more information, see "Security models" in *Manage PAS for OpenEdge*.

New HealthScanner views and custom view support for the OpenEdge HealthScanner

The OpenEdge HealthScanner added two new data views that can be accessed with the following URLs on a running PAS for OpenEdge instance with HealthScanner enabled:

- `http://hostname:8899/health?view=config`
- `http://hostname:8899/health?view=dataset`

These views provide server health information in an ABL dataset-friendly format.

For more information, see "Access HealthScanner data" topic in *Manage PAS for OpenEdge*.

Additionally, OpenEdge 12.2 offers the ability for system administrators and DevOps engineers to write their own custom HealthScanner views for monitoring their PAS for OpenEdge instances.

For more information, see "Write a custom HealthScanner view" in *Manage PAS for OpenEdge*.

Swagger enabled by default on development instances

In OpenEdge 12.2, the Swagger REST API monitoring and testing tool is enabled by default when OpenEdge Manager is deployed to the instance.

For more information, see "Use Swagger UI to explore management REST APIs" in *Manage PAS for OpenEdge*.

PAS for OpenEdge Lite

OpenEdge 12.2 offers a new licensing option for users who want to use PAS for OpenEdge in production.

For more information, see "PAS for OpenEdge Lite" in *Learn About Progress Application Server for OpenEdge*.

OpenEdge Adapters

The following updates were made to the adapter components in OpenEdge 12.2:

Progress OpenEdge JMS Adapter

The Progress OpenEdge JMS Adapter is a generic adapter for Java Messenger Service (JMS) providers that supports any JMS-1.1 compliant provider such as ArtemisMQ, WebSphereMQ, SonicMQ, and others. The Progress OpenEdge JMS Adapter is available as a separate download from the OpenEdge installation. It can be installed on top of an existing OpenEdge 12.2 installation using the same `$DLC` location, or as a standalone service.

For more information, see "Learn about the Progress OpenEdge JMS Adapter" in *Use the Progress OpenEdge JMS Adapter*.

OpenEdge Adapter for Sonic ESB

The OpenEdge Adapter for Sonic ESB allows OpenEdge ABL procedures to be incorporated into ESB processes on the Sonic Enterprise Service Bus (ESB).

For more information, see "Learn about the OpenEdge Adapter for Sonic ESB" in *Use OpenEdge Adapter for Sonic ESB*.

OpenEdge Management

The following updates were made in OpenEdge Management for OpenEdge 12.2:

Improved monitoring of PAS for OpenEdge requests

OpenEdge 12.2 offers improved visibility into active ABL application requests running on PAS for OpenEdge from OpenEdge Management. This feature can help DevOps engineers troubleshoot PAS for OpenEdge performance and operations. A DevOps engineer can look at active ABL application requests to identify, for example, the following information:

- The time the request started
- The length of time that the request took to execute
- The user who sent the request
- The name of the procedure that is running
- The request URL

For more information, see "View OpenEdge ABL application requests" in *Manage Progress Application Server (PAS) for OpenEdge with OpenEdge Management*.

Configure Safe User ID using OpenEdge Management

An option to configure Safe User ID for a multi-tenant database using OpenEdge Management was added in OpenEdge 12.2.

For more information, see "Configure Safe User ID in a multi-tenant database" in *Introduction to Database Multi-Tenancy*.

For more information about Safe User ID, see the "Overview of Safe User ID" in *Learn about Protecting Tenant Data*.

-dbnotifytime option for OpenEdge Management

Support was added for the Database Notification Time (`-dbnotifytime`) option in OpenEdge Management. The `-dbnotifytime` option specifies how often a client polls the database for notifications.

For more information, see "Advanced database configuration propertiesin" *Configure OpenEdge Components with OpenEdge Management*.

DBPOLICYUTIL

A new utility DBPOLICYUTIL was introduced that defines the security policy for a database or a database tenant.

For more information, see "DBPOLICYUTIL utility" in *Manage the OpenEdge Database*.

OpenEdge SQL

OpenEdge SQL offers the following new features in OpenEdge 12.2:

ISO functionality

OpenEdge SQL now supports the ISO `WEEK ()`, `DAY ()` and `YEAR ()` functions for a given input date.

- ISODAY
- ISOWEEK
- ISOYEAR

These topics can be found in the *OpenEdge SQL Reference*.

SQLWidthUpdate

Autonomous Schema Update (ASU) helps resolve the SQL width problem by updating the SQL column width in schema automatically when data exists in columns that are more than their defined width. If the value of the `-SQLWidthUpdate` parameter is set to ON during startup, the Authorized Data Truncation (ADT) feature is also enabled and defaults to OUTPUT irrespective of whether ADT was enabled during startup, unless specified otherwise using the connection URL.

For more information, see "Autonomous schema update" in *OpenEdge Data Management: SQL Development*.

OpenEdge Pro2 6.1

Pro2 Release 6.1 offers new features and enhancements that increase usability and management.

Database schema changes in Pro2 6.1

The `repl.db` schema remains unchanged from 6.0. However, the 5.5 and 6.0 dashboard database, `ReplPerf.db`, has been deprecated in favor of a new database labeled `pro2`. This database resides only within the Pro2 installation folder and now holds configuration information as well as newer performance metrics.

Installation updates

The Pro2 web application is now deployed using `.paar` files in place of `.war` files. The newer method eliminates installation issues encountered with different OpenEdge versions.

Enterprise View dashboard

The Pro2 Enterprise View is a centralized dashboard that aggregates mission-critical data, reports, and metrics across all your Pro2 instances. From the **Enterprise View** dashboard, you can quickly access each of your Pro2 instances to manage and act on any reports or replications from the instance. The **Enterprise** dashboard utilizes a new database labeled Pro2Ent and does not require connectivity to any source or target databases.

For more information, see "Pro2 Enterprise View" in *Use Pro2*.

Job Scheduler

The Pro2 Job Scheduler allows users to optionally discontinue the use of operating system-specific scheduler utilities like the Windows Task Scheduler. The Pro2 Job Scheduler works across all OpenEdge supported operating system infrastructures and provides a common interface for dealing with complex scheduling requirements.

For more information, see "How to schedule jobs in Pro2" in *Use Pro2*.

Status Check improvements

Status check functionality was enhanced to monitor split threads and Enterprise Push program details.

Web user interface enhancements

- The configuration for **LAN/WAN** was moved to the **Settings** Menu (**Menu** > **Settings** > **LAN/Wan**). This eliminates the perpetual prompting for this information in the **New/Edit replication** wizard screens.
- **CDC/Trigger update** is now integrated with the **Select Source** page in the **Manage Replication** wizard.
- All watch-box detail grids were modified to include **Clear All** functionality.
- Instance Dashboard improvements:

  The dashboard and menu structure were modified to provide consistency with the **Enterprise View** dashboard. Additional performance graphs and CDC statistics are now available on the dashboard.

- Safari and Internet Explorer are now included in the list of supported browsers.

De-supported and deprecated features in OpenEdge 12.2

Removal of SonicMQ Adapter configuration in OpenEdge Management

Support for the SonicMQ Adapter configuration in OpenEdge Management was removed in OpenEdge 12.2 because the SonicMQ product, which is now named Aurea Messenger, does not support Java 11. OpenEdge 12.2 requires Java 11.

Older configurations of the SonicMQ adapter (such as for OpenEdge versions 11.7, 12.0, and 12.1) are still supported in OpenEdge Management 12.2.

For more information, see "Configure SonicMQ Adapters" in *Configure OpenEdge Components with OpenEdge Management*.

Removal of the evaluation version of Telerik UI Winforms Control

The Telerik UI Winforms Control evaluation version is not included in the OpenEdge 12.2 installer. If you want to use Telerik UI Winforms with the Progress Developer Studio for OpenEdge, download it from https://www.telerik.com.

Other de-supported and deprecated features in OpenEdge 12.2

For more information about other de-supported and deprecated features in OpenEdge 12.2, see the OpenEdge 12 Product Availability Guide.

5

What's New in OpenEdge 12.1

OpenEdge 12.1 focuses on the following themes:

- High availability on page 56
- Performance on page 57
- Security on page 58
- Developer experience on page 59
- Miscellaneous on page 60

The following topics summarize the new features available for each of these themes.

Important: New Relic plugins for PAS for OpenEdge and the OpenEdge Database.

As of June 2021, OpenEdge discontinued support for the NewRelic plugins for PAS for OpenEdge and the OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to the view plugin dashboard. This limitation is a result of the dropped support by New Relic of the OpenEdge plugins as of Wednesday, June 16th, 2021. The official announcement is available from https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

For details, see the following topics:

- High availability
- Performance
- Security
- Developer experience

- Miscellaneous

High availability

Continuing to address the need for high availability, OpenEdge version 12.1 delivers a range of enhancements and functionality to the OpenEdge database. These features include:

- **Replication properties validation utility**—The DSRUTIL command line utility enables database administrators to validate the properties of replication-enabled source and target databases as well as confirm the status of target database brokers.

  With this feature you can:

  - More readily identify databases within your replication environment
  - Determine the operational status of brokers within your replication environment

  For more information, see OpenEdge Replication DSRUTIL Properties Validation utility.

- **Enhancement to replication status in virtual system tables (VSTs)**—New fields have been added to the _Repl-Server field descriptions and _DbStatus VST field descriptions, exposing additional information required for managing application failover.

  These enhancements allow you to readily identify the status of databases within your replication environment.

- **Modify database parameters online**—The number of parameters that can be modified while the database remains online has increased. Parameters may be changed online through new menu options in PROMON or through VSTs. Support to modify the encryption and security cache sizes (`-ecsize` & `-secsize`) online has also been added to `PROUTIL INCREASETO`. This feature enables you to modify database parameters online and eliminate disruption to users by ensuring continual database availability.

  For more information, see Modify database startup parameters online using PROMON.

- **Allow non-structural schema fields to be changed online**—Database administrators and developers can now change non-structural schema fields online and while the database is being accessed by other users.

  This feature:

  - Does not require an exclusive schema lock to modify non-structural fields
  - Ensures the high availability of the database by enabling users to remain connected while changes are made

  For more information, see Change non-structural fields of _File and _Field online.

- **Allow Integer field to become Int64 online**—You can now change the datatype of a field in an application table from Int to Int64 while the database remains online.

  With this feature you can:

  - Modify the datatype field without an exclusive schema lock
  - Ensure high availability of the database

  For more information, see Change Int to Int64 online.

- **Refresh agents in an ABL application**—The `refreshagents` command terminates all sessions within a multi-session agent so that updates to an ABL application can be made in real-time. The `refreshagents` feature is available as a TCMAN command as well as a REST and OEJMX API.

  The `refreshagents` command allows online schema updates to an application which provides high availability for a PAS for OpenEdge instance. The command may also improve memory leaks in an ABL application.

  For more information, see Refresh agents in an ABL application.

- **Deferred logging in PAS for OpenEdge**—Deferred logging in PAS for OpenEdge opens an in-memory buffer to record stack trace information immediately preceding a multi-session agent crash. It also allows system administrators to run an on-demand monitoring check on an instance via an API.

  Deferred logging helps system administrators understand the cause for an agent crash by providing additional logging information about a crash. Deferred logging also helps performance by only recording a small amount of information that is relevant to an agent crash.

  For more information, see Use deferred logging in PAS for OpenEdge.

- **Get current request information**—The `GET-ACTIVE-REQUESTS-ON-SERVER` API now captures additional details about currently active requests. These additional details provide more visibility into the requests being processed by a PAS for OpenEdge instance. The enhancements to the API better assist system administrators and Technical Support to troubleshoot performance and availability issues with PAS for OpenEdge instances.

  For more information, see Get current request information.

Performance

New database performance enhancements optimize various parameters and capacities as well as provide new functionality for database administrators. These features include:

- **Extend and mark variable-length extents as fixed**—Database administrators can now extend and mark variable-length extents as fixed as well as create and extend new variable-length extents.

  This feature allows you to:

  - Extend the size of an existing variable-length extent before converting it to fixed-length when adding a new variable-length extent to a storage area
  - Extend the size of an existing variable-length extent to a specific pre-grown size whether it is empty or already contains data
  - Extend the size of an existing variable-length extent to a specific pre-grown size and impose a maximum size cap
  - Create a new variable-length extent pre-extended to a specific size in a new or existing area
  - Create a new pre-extended variable-length extent with an imposed maximum size in a new or existing area

  For more information, see Extend variable-length database extents.

- **Limit on the number of unique shared sequences increased**—The maximum number of unique shared sequences in an OpenEdge database is increased to 32,000 regardless of block size.

  This feature improves the performance of the database in a production environment.

  For more information, see Number and size of sequences.

- **Server-side join processing for dynamic queries**—Certain queries that join multiple tables are now resolved on the database server rather than the client. Because fewer records are returned to the client, database performance is improved. In this release, support for server-side joins is expanded to include dynamic queries that are `FORWARD-ONLY`, with `NO-LOCK`.

  For more information on server-side joins for the latest OpenEdge version see *Server-side join processing*.

- **Database startup defaults changed**—Various startup parameter defaults are updated to improve the overall performance of the OpenEdge database. The impacted parameters are:

  - `-aibufs`
  - `-bibufs`
  - `-lruskips`
  - `-lru2skips`
  - `-pica`
  - `-prefetchDelay`
  - `-prefetchFactor`
  - `-prefetchNumRecs`
  - `-prefetchPriority`
  - `-Mm`
  - `-Mxs`

For more information, see Database startup parameter default updates

Security

With continued security improvements, OpenEdge strives to ensure that application functions such as authentication and logging processes offer greater protection and easier debugging. These features include:

- **Tenancy logging for PAS for OpenEdge**—`Sec.TenancyTrace` is a new log entry type for the PAS for OpenEdge agent log file. This improvement in logging helps in troubleshooting an application's management of database user IDs that control access to tenant data.

  For more information, see Tenancy logging.

- **Client authentication support**—Client authentication is now supported for ABL clients connecting to PAS for OpenEdge over APSV and for SOAP, REST, WEB, and static file access via PAS for OpenEdge. The functionality strengthens authentication by requiring that ABL clients authenticate themselves to PAS for OpenEdge.

  For more information, see "Learn about Client Authentication" in *Manage Progress Application Server (PAS) for OpenEdge*.

Developer experience

To support and enable the use of emerging technologies, the following changes were made to the OpenEdge stack:

- **Java Development Kit (JDK)/Java Runtime Environment (JRE)**—In keeping with recent changes to Oracle's licensing of Java, the Java Development Kit (JDK) and Java Runtime Environment (JRE) are no longer included with the OpenEdge installer and are now considered prerequisites for installation. You can:

  - Continue to use the Oracle JDK and JRE
  - Use an open-source JDK and JRE (AdoptOpenJDK is certified)
  - Easily upgrade your version of the JDK and JRE for security upgrades or bug fixes
  - Easily change your path to the JDK and JRE if required

  Depending on the option that you choose to install, you may be prompted to provide the location of your own local copy of the JDK and JRE. If you do not already have the JDK installed, you need to complete this installation before you can complete the installation of this OpenEdge release.

  New to 12.1 is the `$DLC/properties/java.properties` file where the `JAVA_HOME` environment variable is set. Any changes made to this file for the JDK and JRE go into effect the next time the Docker image is started.

  For information on JDK and JRE support requirements, see *OpenEdge 12 Platform and Product Availability Guide*.

- **Run PAS for OpenEdge in a Docker container**—The recommended approach for providing a JDK to a PAS for OpenEdge Docker container is the Docker sidecar design pattern. The JDK is deployed in a separate (sidecar) Docker container. The JDK container is made available to the PAS for OpenEdge container via a volume share.

  The JDK image in the Docker container must be a version supported by PAS for OpenEdge. The image can be from Docker hub or a custom image provided by you. How you attach the sidecar container to the PAS for OpenEdge container depends on how you created the container (Docker run, Docker Compose, or Minikube). A sample script for each approach is bundled in this release. These scripts can be modified per your needs.

  For more information, see Learn About PAS for OpenEdge in a Docker Container.

- **Unified logging for PAS for OpenEdge Containers**—Extending its commitment to cloud computing and scalability, OpenEdge now supports using Fluent Bit with PAS for OpenEdge.

  This release provides:

  - An improved process for customizing the configuration of Fluent Bit
  - The default output of Fluent Bit is to the standard output (which is the best practice for docker containers)

  For more information, see Learn About Unified Logging.

- **File holding the schema will be searched in PROPATH**—To improve the usability and efficiency of the Developer Studio, the schema file for Business Entities is now defined in PROPATH.

  The advantages of adding the schema file to PROPATH are:

  - PROPATH maintains the list of all the paths required for a project. If a schema file is defined in any of the paths, Developer Studio automatically searches the path and associates the schema file with the

project thus eliminating the need to copy the file to the workspace and specify its relative or absolute path.

- During catalog generation, the relative path specified in the Business Entity is resolved against PROPATH entries.
- The same Business Entity can be used across development environments by modifying the PROPATH entries to include the relative path of the entity.
- If you select a schema file from a directory in your PROPATH, the relative path to the directory is displayed in the Schema File field.

To make the Progress Developer Studio for OpenEdge easier to use and faster to develop with, Schema File for defined Business Entities are added to the PROPATH.

For more information, see Select a schema file page.

Miscellaneous

- **Windows Install package for Client Networking**—A separate Windows Install package is now available to exclusively install Client Networking. In addition, Client Networking continues to be available as part of the OpenEdge installer.

  For more information, see Use the Microsoft Windows Installer Package for Client Networking.

- **Upgraded Aurea Sonic client JARs**—Aurea Sonic client JARs are upgraded to support the latest Aurea Sonic 2018 release.

6

What's New in OpenEdge 12.0

OpenEdge Release 12.0 includes several product updates that are designed to achieve the following performance goals:

- Continuous operation—The goal of continuous operations is to keep your system running 24x7. This requires resiliency to failure, the ability to perform online maintenance while the system is running, and eliminating the need to bring the system down to roll out business logic or schema updates.

  To enhance its support of continuous operations, OpenEdge 12.0 introduces the following:

  - HealthScanner: a low-overhead server instance and system monitoring tool that system administrators can use to determine whether any performance issues exist. This gives you the opportunity to correct the issue, using a load balancer or a tool like Nagios, before an interruption of service occurs.
  - Improved Stop Agent and Idle Agent support: the Stop Agent API has been updated to allow all requests to normally finish up to a specific timeout before issuing a STOP, and waiting for all tasks to finish, before taking the agent down. The idleAgentTimeout property has been enhanced to accept these values so that it can be tuned for your application. Therefore, if the HealthScanner notices an agent is becoming unhealthy due to increasing memory usage, a tool such as Nagios can replace that agent before users are affected.
  - OpenEdge Replication after-image (AI) file streaming: in the event of a failure, replication stream blocks are immediately transferred to all target machines, not queued on the source machine for transfer. This leads to improved replication performance with minimal loss of data.
  - Replication agents run during transition: from a client perspective, the transition to the new database broker is smooth. The client no longer needs to disconnect and then connect to the new database broker—it happens automatically.
  - Online operations: several operations can now be performed without requiring OpenEdge Database to be shut down, such as before-image (BI) space reclamation, database log file rotation, schema changes, JVM configuration, index builds, and more.

- Service and data integration—Going forward, more and more applications are being assembled from existing services. Service architectures lend themselves to reuse, facilitate agility, and are foundational for web and mobile needs. Additionally, the modernization of existing applications towards a service architecture is an ongoing evolution, and OpenEdge 12.0 addresses this evolution by introducing Swagger UI support for viewing REST APIs. Swagger allows system administrators to explore the underlying management APIs for a running PAS for OpenEdge instance.
- Security—PAS for OpenEdge includes support for OAuth2 and SAML, cryptography updates, Spring Security updates, and more.
- Performance and scalability—In previous releases, database connection requests were handled one-at-a-time. If a request is blocked, it is put on hold until the remaining requests are processed. Only then does processing return to the blocked request, where the block may have been resolved for some time before processing becomes available again. In Release 12.0, OpenEdge Database introduces multi-threading, so if a request is blocked, processing can continue as soon as the block is resolved. In addition, with OpenEdge ABL performance enhancements, shared memory is eliminated as a single point of failure.
- Improved developer experience—The developer experience pertains to the coding, debugging, testing, and deployment of an application. To be productive, developers need to be equipped with a complete toolchain, from algorithms and backend services, to user interfaces and advanced components.

  In Release 12.0, OpenEdge provides the following features to improve the developer experience:

  - Docker support: you can now run PAS for OpenEdge in a Docker container, enabling continuous integration and continuous deployment (CI/CD) of your OpenEdge applications.
  - ABL enhancements: new memory stream classes, the ability to resize a fixed indeterminate array, STOP condition improvements, new server-side join processing,and more.
  - Progress Developer Studio for OpenEdge updates: Performance Profiler enhancements, new ABL editor options, support for importing and exporting workspace settings, and bundled tool updates including a new code analyzer for ABL.

For more information, see the following topics:

- Progress Application Server for OpenEdge on page 63
- SQL on page 65
- Database on page 66
- OpenEdge Management on page 69
- ABL on page 70
- Progress Developer Studio for OpenEdge on page 73
- DataServers on page 74

Important: New Relic plugins for PAS for OpenEdge and the OpenEdge Database.

As of June 2021, OpenEdge discontinued support for the NewRelic plugins for PAS for OpenEdge and the OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to the view plugin dashboard. This limitation is a result of the dropped support by New Relic of the OpenEdge plugins as of Wednesday, June 16th, 2021. The official announcement is available from https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

For details, see the following topics:

- Progress Application Server for OpenEdge
- SQL
- Database
- OpenEdge Management
- ABL
- Progress Developer Studio for OpenEdge
- DataServers

Progress Application Server for OpenEdge

The following topics describe the continuous operation features that are added to Progress Application Server (PAS) for OpenEdge in Release 12.0:

- HealthScanner on page 63
- Docker Container support on page 64
- Unified logging for PAS for OpenEdge container on page 64
- Swagger UI support for PASOE management APIs on page 64
- Improved agent stop on page 64
- Server-side ABL performance profiling on page 64
- Security updates on page 65

HealthScanner

Designed for system administrators, the PAS for OpenEdge HealthScanner can detect potential problems with a PAS for OpenEdge instance so that the instance can be taken out of service before a failure occurs. The HealthScanner continuously monitors key system health metrics, such as CPU health, REST ping health, HTTP request health, and disk space and memory health, and generates an overall score. If the score falls below a certain threshold, a server instance can be taken out of service by an elastic load balancer and replaced before any disruption of service is incurred.

For more information, see 'Use the OpenEdge HealthScanner' in *Manage Progress Application Server for OpenEdge*.

Docker Container support

In Release 12.0, PAS for OpenEdge is available in a Docker container. Docker containers allow developers to develop, deploy, and run scalable applications in an isolated environment. Using containers to enable faster and easier development and deployment.

Unified logging for PAS for OpenEdge container

When you run PAS for OpenEdge in Docker containers, sorting through all the available logs throughout the file systems in each container is time consuming and inefficient. To simplify log monitoring, OpenEdge includes the ability to use EFK (Elastic Fluentd Kibana) stack logging to create a unified logging structure. A unified logging structure allows you to access log files across all of your containers from a single location. EFK stack logging uses Elasticsearch, Fluent Bit, and Kibana to collect logs from multiple containers and make them available for display.

For more information, see “Learn About Unified Logging” in *Use Unified Logging with a Progress Application Server for OpenEdge Container*.

Swagger UI support for PASOE management APIs

System administrators can use Swagger UI to obtain details about the underlying management REST APIs for a running PAS for OpenEdge instance. Details about the instance are extracted from annotations that are embedded within the APIs, ensuring that the generated documentation is always current. Using this feature, you can now view performance metrics, identify deployed applications, stop agents, and do much more using this interactive documentation tool.

Improved agent stop

OpenEdge 12.0 introduces two agent stop improvements:

- The Stop Agent API has been updated to allow all requests to normally finish up to a specific timeout before issuing a `STOP`, and waiting for all tasks to finish, before taking the agent down.

  For more information, see “Use Swagger UI to explore management REST APIs” in *Manage Progress Application Server for OpenEdge*.

- Similarly, the `idleAgentTimeout` property has been enhanced to accept timeout values that can be tuned for your application.

  See “Configure tuning parameters and process limits” in *Manage Progress Application Server for OpenEdge with OpenEdge Management*

Server-side ABL performance profiling

System administrators and developers can use server-side ABL performance profiling to gather and monitor run-time performance data of ABL applications running in a PAS for OpenEdge instance.

For more information, see “Use Server-Side ABL Performance Profiling” in *Manage Progress Application Server for OpenEdge*.

Security updates

OpenEdge 12.0 includes the following security updates.

Spring Security updates

In OpenEdge 12.0, `bcrypt` is the default and only supported hashing algorithm for encrypting passwords using the `GENSPRINGPWD` utility. Previously supported algorithms–`sha256`, `sha512`, and `sha1`–are discontinued because of their known vulnerability.

For details, see “Generate encrypted passwords with GENSPRINGPWD” in *Administer Progress Application Server for OpenEdge*.

OpenSSL

In OpenEdge 12.0, transport layer security has been strengthened by upgrading OpenSSL to 1.1.1.

For more information, see “Supported protocols, ciphers, and certificates for OpenEdge clients and servers” in *Introduction to Security and Auditing*.

SECPROP utility

OpenEdge Release 12.0 introduces the SECPROP utility, which greatly simplifies configuring and managing the security properties for a PAS for OpenEdge web application that are defined in the `oeablSecurity.properties` file.

For more information, see “SECPROP” in *Administer Progress Application Server for OpenEdge*.

Progress-supplied root certificates

OpenEdge provides several root digital (public-key) certificates from trusted Certificate Authorities for clients and servers that support TLS network connections. These root certificates are shipped as part of OpenEdge and are located in the `certs` directory within the OpenEdge installation directory.

In release 12.0, the set of root certificates has been updated to include those from the following vendors:

- DigiCert
- Entrust
- Go Daddy
- GlobalSign
- GeoTrust
- Thawte
- Symantec

For information about managing root certificates in the OpenEdge keystore, see "Manage certificate stores for OpenEdge clients and servers" in *Manage OpenEdge Keys and Certificates*.

SQL

OpenEdge 12.0 includes the following new features in OpenEdge SQL:

- Online schema changes on page 66

- Online JVM configuration on page 66
- Troubleshoot information added to Protrace report on page 66
- Autonomous Update Statistics on page 66

Online schema changes

With online schema changes made in OpenEdge SQL 12.0, multiple database operations can be performed concurrently without interfering with each other. The benefit of online schema changes is that OpenEdge SQL objects do not have an exclusive lock and are accessible to other queries and operations as a result.

For details, see “Online schema changes” in *Develop SQL for OpenEdge*.

Online JVM configuration

For an existing JVM, you can now reconfigure the JVM with new parameters and put them in effect without manually shutting down and restarting the database. Instead, the OpenEdge SQL Server waits until there are no database connections, and then automatically shuts down the OpenEdge SQL Server process and restarts the JVM with the new parameters. If a JVM is not already created in a running OpenEdge SQL Server process, then changes to the JVM parameters do not force the SQL Server process to restart.

For more information, see “Configure JVM and specify the parameters” in *Develop SQL for OpenEdge*.

Troubleshoot information added to Protrace report

The Protrace crash report helps to quickly identify and diagnose the root cause of a SQL crash for a query that’s terminated abnormally. In OpenEdge SQL 12.0, the relevant SQL query is now captured in the `protrace.`*`pid`* file along with the additional information useful for debugging.

For more information, see “Troubleshoot a SQL Crash” in *Develop SQL for OpenEdge*.

Autonomous Update Statistics

OpenEdge SQL 12.0 provides a mechanism for identifying whether OpenEdge SQL statistics are stale and need to be updated for tables that are involved in `SELECT` statements. With the autonomous update statistics (AUS) service, you can now automatically update and collect statistics during a maintenance window. AUS offers the benefit of using up-to-date statistics while saving the time and effort involved in manually updating statistics.

For details, see “Autonomous update statistics” in *Develop SQL for OpenEdge*.

Database

The following features have been added to OpenEdge Database in Release 12.0:

- AI file streaming on page 67
- Multi-threaded server on page 67

- Reduced locking delays in buffer hash table on page 67
- Modifiable VSTs in replication target on page 67
- Appropriate file permissions for conversion utilities on page 68
- Sequences all 64-bit on page 68
- Database startup parameter changes on page 68
- Large file support on page 68
- Synchronous commit mode no longer supported on page 68
- Improved transition control on page 68
- Encryption changes for binary dump and load on page 69

AI file streaming

DBAs want their disaster recovery system to have the most up to date database changes so that in the event of a failure, recovery can occur quickly and with minimal data loss. OpenEdge Release 12.0 introduces Replication AI Streaming to meet this requirement. To minimize the loss of AI data queued on a source, OpenEdge Replication AI Streaming now transmits AI blocks to the targets as they are generated.

For more information, see “Replication AI Streaming overview” in *Replication AI Streaming*.

Multi-threaded server

The OpenEdge Database server is enhanced in Release 12.0 to use a multi-threaded model in order to process remote client requests concurrently. This capability improves database performance and provides for horizontal scalability.

This capability improves database performance and provides for horizontal scalability.

See “Threaded Database Server Functionality and Benefits” in *Introduction to Threaded OpenEdge Databases* for details.

Reduced locking delays in buffer hash table

Changes in 12.0 work to reduce lock delays in the Buffer Hash Table. Enhancements to small table scans enable find next operations to reference the index block last visited in the buffer pool without the need for hash table lookups, while the Buffer Hash Latch Factor parameter (`-hashLatchFactor`) enables database administrators to tailor the number of latches allocated for the Buffer pool Hash Table (BHT) entries.

See “Buffer Hash Latch Factor (-hashLatchFactor)” in *Startup Command and Parameter Reference*.

Modifiable VSTs in replication target

To ease virtual system table (VST) updating on replication targets, the following modifiable VST tables and fields are now allowed on replication targets:

- `_Connect`
- `_MyConnect`

- _Database-Feature
- _DbParams

For more information, see “Modifiable VSTs on Replication Targets” in *Use Database Replication*.

Appropriate file permissions for conversion utilities

In OpenEdge Database 12.0, permissions for DBUTILS files on Windows NT64, AIX64, and Linux64 platforms each differ for the conversion process. Database administrators need to verify their permissions in the DBUTILS directory before converting.

For details, see “Converting an OpenEdge Release 10 database to OpenEdge Release 12” in *Manage the OpenEdge Database*.

Sequences all 64-bit

In Release 12, 64-bit sequences are enabled by default in newly created databases.

For details, see “Number and size of sequences” in *Manage the OpeEdge Database*.

Database startup parameter changes

The default values for several OpenEdge Database startup parameters have changed in Release 12.0.

For details, see “Database server performance parameters” in *Startup Command and Parameter Reference*.

Large file support

Large file support is enabled by default in Release 12 for particular license levels. Opening a database on systems running with an enterprise license enables large file support if it is not enabled already. Opening a database on systems running with a sub-enterprise license (Workgroup and Personal) will disable large file support, if it is enabled.

For details, see “Enable large files” in *Introduction to the Database*.

Synchronous commit mode no longer supported

Synchronous commit mode is not supported in OpenEdge Database 12.0. In prior releases, you had the option of synchronous or asynchronous commits from your source database to your target. Starting in Release 12.0, only asynchronous commit mode is supported.

For more information, see “Asynchronous replication” in *Manage the OpenEdge Database*.

Improved transition control

To ease the configuration of transition and transition failover properties, OpenEdge Database Release 12 offers two new parameters that may be set without the need to change the properties file:

- dr-transition-to-agents

- `transition-to-agents`

For details about these parameters, see “Transition properties summary” in *Use Database Replication*.

Encryption changes for binary dump and load

Two additional ciphers have been added in support of binary dump and load: `Cipher 11` and `Cipher 12`. Details about these ciphers can be found using the `EPOLICY INFO` qualifier to the PROUTL utility.

For more information, see “Binary dump and load of data” in *Manage the OpenEdge Database*.

OpenEdge Management

In Release 12.0, OpenEdge Management includes the following product updates:

- Simplified PAS for OpenEdge creation wizard on page 69
- Simplified PAS for OpenEdge configuration on page 69
- OpenEdge Management now runs independently of AdminServer on page 69

Simplified PAS for OpenEdge creation wizard

The new PAS for OpenEdge creation wizard has been enhanced to include context based fields. Also, a new security model has been added to the page, which enables you to choose to create a PAS for OpenEdge instance in either production or development mode.

For details, see “Create a PAS for OpenEdge instance” in *Manage Progress Application Server for OpenEdge with OpenEdge Management*.

Simplified PAS for OpenEdge configuration

The PAS for OpenEdge configuration page has been made easier to use.

For details, see “Configure a PAS for OpenEdge instance with OpenEdge Management” in *Manage Progress Application Server for OpenEdge with OpenEdge Management*.

OpenEdge Management now runs independently of AdminServer

Beginning with OpenEdge Release 12.0, OpenEdge Management is hosted on the Progress Application Server. Progress Application Server is a Tomcat Web server with Progress-specific configuration that enables OpenEdge Management to run independently of the AdminServer. This change also enables you to manage multiple local AdminServer instances in addition to remote AdminServer instances.

OEM runs as a stand-alone web application and it is not capable of managing any local or remote OE resources/system resources with out the AdminServer (local network resources are an exception). All the communication to the local/remote resources happens through the AdminServer. In order to manage and monitor any resources the AdminServer needs to be started.

For details, see “OpenEdge Management AdminServer Configuration Upgrade” in *Upgrade to OpenEdge 12*.

ABL

OpenEdge ABL (Advanced Business Language) provides the following new features in OpenEdge 12.0:

- New memory stream classes on page 70
- Indeterminate array resizing improvement on page 70
- Error and stop object enhancements for a Java open client on page 70
- STOP condition processing on page 71
- Behavior change for empty FIELDS list on page 71
- Client-principal updates on page 72
- New server-side join processing on page 72
- R-code updates on page 72

New memory stream classes

OpenEdge Release 12.0 is enhanced to include two new memory stream classes, `Progress.IO.MemoryOutputStream` and `Progress.IO.MemoryInputStream`, plus a new enumeration class that supports the MemoryOutputStream object's ByteOrder property.

Apart from providing memory stream capability, these new classes also help to encapsulate the MEMPTR data type, which makes it easier to perform MEMPTR operations.

For more information, see “Progress.IO.MemoryOutputStream class” and “Progress.IO.MemoryInputStream class” in *ABL Reference*.

Indeterminate array resizing improvement

OpenEdge Release 12.0 adds the ability in ABL to resize a fixed indeterminate array. In previous releases you had to unfix the size before setting it to a new fixed size, which resulted in the loss of all data in the array. To prevent this you had to save all data in the array, unfix the size, set the new size, and then copy the saved data to the new resized array. With this release the indeterminate array can be resized without the loss of data. You simply call the `EXTENT` statement to resize it.

For more information, see “EXTENT statement” in *ABL Reference*.

Error and stop object enhancements for a Java open client

OpenEdge Release 12.0 adds new classes and methods for passing error and stop objects to a Java open client. These new classes and methods enable you to obtain more detail on the type of error that occurred on the server. New and modified classes and methods include:

Class name	Inherits from	Methods/properties
RunTime4GLErrorException	com.progress.open4gl.RunTime4GLException	getABLCallStack(), getNumMessages(), getSeverity(), getMessage(msgIx), getMessageNum(msgIx)
RunTime4GLStopException	com.progress.open4gl.RunTime4GLException	getABLCallStack()
RunTime4GLAppErrorException	com.progress.open4gl.RunTime4GLErrorException	No methods or properties of its own
RunTime4GLStopErrorException	com.progress.open4gl.RunTime4GLErrorException	No methods or properties of its own
RunTime4GLStopAfterException	com.progress.open4gl.RunTime4GLStopException	No methods or properties of its own
RunTime4GLLockConflictException	com.progress.open4gl.RunTime4GLStopException	getTableName(), getUser(), getDevice()

For more information see the Javadoc, available at `OpenEdge-install-directory/java/doc/com/progress/open4gl/package-summary.html`.

STOP condition processing

In OpenEdge Release 12.0 `STOP` conditions behave more like errors. When a `STOP` condition is raised, the AVM throws an appropriate stop object, executes any appropriate `CATCH` blocks in response, and executes `FINALLY` blocks. Even if the `STOP` condition is not handled in the current block, the AVM runs a local `FINALLY` block. There are still a few, rare conditions that cause `STOP` to bubble up to the top of a transaction block, or out to a layer that does not access the database, and not abide by any `ON STOP` or relevant `CATCH` blocks along the way. For those conditions, `FINALLY` blocks along that path will not run either.

In 11.7 you had to explicitly enable this behavior by using the `(-catchStop)` startup parameter. This is now the default in 12.0. To revert to the previous behavior, use `-catchStop 0`.

For more information, see “CATCH Blocks” and “FINALLY Blocks” in *ABL Error Handling*.

For more information about the `-catchStop` parameter, see “Catch STOP (-catchStop)” in *Startup Command and Parameter Reference*.

Behavior change for empty FIELDS list

Prior to Release 12.0 specifying an empty `FIELDS` list (`FIELDS()`) during record retrieval would return the entire record. Now in 12.0 an empty `FIELDS` list returns no fields. However the ROWID value for a specified record is available using the `ROWID` function. To revert to the previous behavior of returning all fields, remove the `FIELDS()` option from the record phrase so that no field list is specified. A `FIELDS` list can be added to a `FOR`, `DO PRESELECT`, `REPEAT PRESELECT`, or `DEFINE QUERY` statement. To mitigate runtime errors related to field lists you can use the `-rereadfields` or `-fldisable` startup parameters.

For more information about the record phrase, see “Record phrase” in *ABL Reference*.

For more information about the `-rereadfields` and `-fldisable` startup parameters, see “-rereadfields” and “-fldisable” in *Startup Command and Parameter Reference*.

Client-principal updates

OpenEdge Release 12.0 introduces the following changes that affect the client-principal object:

- A more secure algorithm is used for sealing a client-principal object. Note that a 12.0 process is **not** able to accept a client-principal object sealed in a prior OpenEdge release.
- The SEAL( ) method now only accepts a clear text domain access code as input.
- Domain access codes stored in an OpenEdge database use a different encoding from previous releases. To support this new encoding the SECURITY-POLICY system handle has two new methods: ENCODE-DOMAIN-ACCESS-CODE( ) and VALIDATE-DOMAIN-ACCESS-CODE( ).

For more information about the SEAL( ) method, see “SEAL( ) method” in *ABL Reference*.

For more information about the SECURITY-POLICY system handle, see “SECURITY-POLICY system handle” in *ABL Reference*.

For more information about the ENCODE-DOMAIN-ACCESS-CODE( ) method, see “ENCODE-DOMAIN-ACCESS-CODE( ) method” in *ABL Reference*.

For more information about the VALIDATE-DOMAIN-ACCESS-CODE( ) method, see “VALIDATE-DOMAIN-ACCESS-CODE( ) method” in *ABL Reference*.

New server-side join processing

OpenEdge Release 12.0 is enhanced to perform server-side joins. Certain queries that join multiple tables are now resolved in the database server rather than the client. As a result, fewer records may be returned to the client leading to faster performance. Server-side join is turned on by default in a multi-threaded database server and only for certain FOR statements containing up to 10 tables in the same logical database. Server-side join cannot be activated in a non-threaded database server.

For 12.0 server-side join is limited to FOR statements that use NO-LOCK.

For more information on server-side join for the current version of OpenEdge, see *Server-side join processing*.

R-code updates

OpenEdge 12 has been upgraded to use a more secure encryption algorithm, resulting in changes to the r-code.

The GENERATE-MD5 option was removed from the COMPILE statement. Prior to 12.0 this option produced a unique MD5 value that was stored in the r-code file. It could be used to determine if an r-code file changed since the previous version of the application. Now in 12.0, a signature value is automatically generated instead using a more secure encryption algorithm. The MD5-VALUE attribute for the RCODE-INFO handle is now obsolete and returns the Unknown (?) value. It is replaced by the new SIGNATURE-VALUE attribute.

By default an object serialized in a pre-12.0 release cannot be deserialized in 12.0. For backward compatibility a new attribute was added to the SECURITY-POLICY system handle called ALLOW-PREV-DESERIALIZATION. Setting the attribute to YES/TRUE allows an application to receive an object using the serialization from the previous OpenEdge major release version.

For more information about the COMPILE statement, see “COMPILE statement” in *ABL Reference*.

For more information about the MD5-VALUE attribute, see “MD5-VALUE attribute” in *ABL Reference*.

For more information about the RCODE-INFO handle, see “RCODE-INFO handle” in *ABL Reference*.

For more information about the SIGNATURE-VALUE attribute, see “SIGNATURE-VALUE attribute” in *ABL Reference*.

For more information about the SECURITY-POLICY system handle, see “SECURITY-POLICY system handle” in *ABL Reference*.

For more information about the ALLOW-PREV-DESERIALIZATION attribute, see “ALLOW-PREV-DESERIALIZATION attribute” in *ABL Reference*.

Progress Developer Studio for OpenEdge

The following features are added to Progress Developer Studio for OpenEdge Release 12.0:

- Performance profiler enhancements on page 73
- New ABL editor options on page 73
- Support for importing and exporting workspace settings on page 73
- Bundled third-party tool updates on page 74
- Telerik UI for WinForms update on page 74

Performance profiler enhancements

The ABL code now has Performance Profiler enhancements that show true Call-Tree data and the PROPATH.

For more information, see “Call Tree tab” and “AVM Information tab” in *Progress Developer Studio for OpenEdge Online Help*.

New ABL editor options

The ABL Editor section in Progress Developer Studio for OpenEdge has been updated in OpenEdge 12.0 to allow you to perform the following tasks:

- **Code formatting**—Progress Developer Studio for OpenEdge now allows you to format the ABL code. The Code Formatter page (**Window** > **Preferences** > **Progress OpenEdge** > **Editor** > **Code Formatter**) of the Editor preferences includes the options in the Control Statements tab.

  See “Editor Code Formatter options” in *Progress Developer Studio for OpenEdge Online Help*.

- **Shortcut to compile an ABL file**—You can now compile an ABL file using the keyboard shortcut CTRL+ALT+C.

  See “Compile ABL code” in *Progress Developer Studio for OpenEdge Online Help*.

Support for importing and exporting workspace settings

In Release 12.0 of Progress Developer Studio for OpenEdge, you can now import and export workspace settings.

For more information, see “Import the Workspace Settings” and “Export the Workspace Settings” in *Progress Developer Studio for OpenEdge Online Help*

Bundled third-party tool updates

In Release 12.0, the following third-party tools that are bundled with Progress Developer Studio for OpenEdge have been updated. For details about all third-party tools packaged with OpenEdge, see *OpenEdge 12 Platform & Product Availability Guide* on the Progress Information Hub at https://docs.progress.com.

- **Code Analyzer for ABL**—Helps to achieve coding best practices and to improve product performance.

  See "Introduction to Code Analyzer for ABL" in *Progress Developer Studio for OpenEdge Online Help* for details.

- **Progress Compilation Tools version update**—Progress Compilation Tools (PCT) is a set of ANT libraries that can be used in the OpenEdge environment to help to automate build generation in OpenEdge applications. In Release 12.0, OpenEdge includes PCT version 210.

- **Infragistics controls update**—OpenEdge release 12.0 uses Infragistics NetAdvantage for .NET. The version of the Infragistics controls has been updated to 18.2.20182.175.

Telerik UI for WinForms update

The version of Telerik UI for WinForms that is bundled with Progress Developer Studio for OpenEdge Release 12.0 has been updated. For details, see *OpenEdge 12 Platform & Product Availability Guide* on the Progress Information Hub at https://docs.progress.com. If you need a different version, then download it directly from Telerik.

DataServers

OpenEdge DataServers Release 12.0 includes the following updates:

- Requirement to build executables on AIX is lifted on page 74
- Dynamic loading of Oracle client library in OpenEdge DataServer is enabled on page 74

Requirement to build executables on AIX is lifted

On supported AIX platforms, building the DataServer executables is no longer required. OpenEdge provides out-of-the-box executables that can be used directly.

For details, see "Building DataServer Executables" in *Use the Oracle Data Server*.

Dynamic loading of Oracle client library in OpenEdge DataServer is enabled

On AIX platforms, building the DataServer executables for use with OpenEdge DataServer for Oracle is no longer required. You can now use the executables provided by Progress directly, such as `_progres`, `_orasrv`, and `_mproapsv`. You need only to set the `ORACLE_HOME` environment variable to the top-level directory of your Oracle installation.

For details, see "Building DataServer Executables" in *Use the Oracle Data Server*.