Progress®



What's New in OpenEdge 12

OpenEdge®

Copyright

© 2019 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.

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Deliver More Than Expected, Icenium, Kendo UI, Kinvey, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. Analytics360, AppServer, BusinessEdge, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, JustAssembly, JustDecompile, JustMock, Kinvey Chat, NativeChat, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, 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.

February 2019

Last updated with new content: Release 12.0

Updated: 2021/06/15

Table of Contents

Preface....7

What's New in OpenEdge 12.0....9

Progress Application Server for OpenEdge....11
HealthScanner....11
Docker Container support....12
Unified logging for PAS for OpenEdge container....12
Swagger UI support for PASOE management APIs....12
Improved agent stop....12
Server-side ABL performance profiling....12
Security updates....13
SQL....14
Online schema changes....14
Online JVM configuration....14
Troubleshoot information added to Protrace report....14
Autonomous Update Statistics....14
Database....15
AI file streaming....15
Multi-threaded server....15
Reduced locking delays in buffer hash table....15
Modifiable VSTs in replication target....16
Appropriate file permissions for conversion utilities....16
Sequences all 64-bit....16
Database startup parameter changes....16
Large file support....16
Synchronous commit mode no longer supported....16
Improved transition control....17
Encryption changes for binary dump and load....17
OpenEdge Management....17
Simplified PAS for OpenEdge creation wizard....17
Simplified PAS for OpenEdge configuration....17
OpenEdge Management now runs independently of AdminServer....18
ABL....18
New memory stream classes....18
Indeterminate array resizing improvement....18
Error and stop object enhancements for a Java open client....19
STOP condition processing....19
Behavior change for empty FIELDS list....20
Client-principal updates....20
New server-side join processing....20

R-code updates....20
Progress Developer Studio for OpenEdge....21
Performance profiler enhancements....21
New ABL editor options....21
Support for importing and exporting workspace settings....22
Bundled third-party tool updates....22
Telerik UI for WinForms update....22
DataServers....22
Requirement to build executables on AIX is lifted....23
Dynamic loading of Oracle client library in OpenEdge DataServer is enabled....23

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 12.0 on page 9

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

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 11
- SQL on page 14
- Database on page 15
- OpenEdge Management on page 17
- ABL on page 18
- Progress Developer Studio for OpenEdge on page 21
- DataServers on page 22

Important: NewRelic plugin for PAS for OpenEdge and NewRelic plugin for OpenEdge Database

As of June 2021, OpenEdge will discontinue support for the NewRelic plugin for PAS for OpenEdge and NewRelic plugin for OpenEdge Database. Customers using NewRelic plugins will not be able to access the plugin pages to configure metrics or to view plugin dashboard. This is because New Relic will no longer support or maintain Plugins as of Wednesday, June 16th, 2021. Here is an update from NewRelic Web Site: https://discuss.newrelic.com/t/new-relic-plugin-eol-wednesday-june-16th-2021/127267.

At present, there are no alternative solutions to the OpenEdge NewRelic plugins.

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 11
- Docker Container support on page 12
- Unified logging for PAS for OpenEdge container on page 12
- Swagger UI support for PASOE management APIs on page 12
- Improved agent stop on page 12
- Server-side ABL performance profiling on page 12
- Security updates on page 13

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 14
- Online JVM configuration on page 14
- Troubleshoot information added to Protrace report on page 14
- Autonomous Update Statistics on page 14

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 15
- Multi-threaded server on page 15
- Reduced locking delays in buffer hash table on page 15
- Modifiable VSTs in replication target on page 16
- Appropriate file permissions for conversion utilities on page 16
- Sequences all 64-bit on page 16
- Database startup parameter changes on page 16
- Large file support on page 16
- Synchronous commit mode no longer supported on page 16
- Improved transition control on page 17
- Encryption changes for binary dump and load on page 17

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 17
- Simplified PAS for OpenEdge configuration on page 17
- OpenEdge Management now runs independently of AdminServer on page 18

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 18
- Indeterminate array resizing improvement on page 18
- Error and stop object enhancements for a Java open client on page 19
- STOP condition processing on page 19
- Behavior change for empty FIELDS list on page 20
- Client-principal updates on page 20
- New server-side join processing on page 20
- R-code updates on page 20

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 21
- New ABL editor options on page 21
- Support for importing and exporting workspace settings on page 22
- Bundled third-party tool updates on page 22
- Telerik UI for WinForms update on page 22

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 23
- Dynamic loading of Oracle client library in OpenEdge DataServer is enabled on page 23

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*.