Progress®



Upgrade to 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

Product version: Progress OpenEdge 12.4

Updated: 2021/06/29

Table of Contents

Overview of Upgrading to OpenEdge 12....9

Convert an OpenEdge Release 11 Database to OpenEdge Release 12...11

Prepare to convert replication-enabled databases....13
Convert replication-enabled databases without target rebase....14

OpenEdge Management AdminServer Configuration Upgrade....17

Development Environment Upgrade Tasks....23

Upgrade an ABL WebApp project....23
Updates to the cryptography used to secure client-principal objects....24
Package an ABL web application ANT project....24
Upgrade ABL Service Projects in a PAS for OpenEdge Instance....24
Upgrade ABL Service projects....25
Create the PAS for OpenEdge instance....26

Upgrade Security to OpenEdge 12....29

Updates to password encryption with GENSPRINGPWD....29
PAS for OpenEdge and OpenEdge Authentication Gateway: Domain configuration....30
Secure communications in OpenEdge Management....31
Upgrade an OpenEdge Authentication Gateway instance configuration....31

Upgrade ABL Applications....33

Migration notes....35

Preface

Purpose

Upgrade to OpenEdge 12 describes the tasks that you may need to perform to upgrade an OpenEdge release 11.7 deployment environment to OpenEdge release 12. Manual configuration is required if you are upgrading to OpenEdge 12 and already have workspaces, instances, projects, and so on, from the following products:

- Progress Application Server (PAS) for OpenEdge
- OpenEdge Database
- OpenEdge Management
- Progress Developer Studio for OpenEdge

Audience

Upgrade to OpenEdge 12 is for ABL developers and database administrators who are responsible for upgrading an OpenEdge development or production environment to Release 12.

Organization

Overview of Upgrading to OpenEdge 12 on page 9

Lists and summarizes the sequence of tasks that you may need to perform to upgrade an OpenEdge Release 11 environment to OpenEdge 12.

Convert an OpenEdge Release 11 Database to OpenEdge Release 12 on page 11

Gives the steps to convert the schema of an OpenEdge Release 11 database to Release 12. This topic also explains how to convert a database that is enabled for Transparent Data Encryption (TDE).

OpenEdge Management AdminServer Configuration Upgrade on page 17

Explains the changes that that are required to the AdminServer configuration of OpenEdge Management.

Development Environment Upgrade Tasks on page 23

Explains the steps required for upgrading the OpenEdge application development environment and Spring Security files to OpenEdge 12.

Upgrade Security to OpenEdge 12 on page 29

Describes important notes regarding security updates made in OpenEdge Release 12, and also describes tasks you may need to complete to upgrade your application environment security to Release 12.

Upgrade ABL Applications on page 33

Explains changes to security in OpenEdge 12 that may require you to upgrade your ABL applications, depending on the specific OpenEdge features that your ABL applications use.

Migration notes on page 35

Provides notes regarding configuration changes made in OpenEdge 12 that may be relevant to components you used in OpenEdge 11.7.

Documentation conventions

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

1

Overview of Upgrading to OpenEdge 12

When you upgrade an OpenEdge Release 11 environment to OpenEdge 12, there are a number of tasks that you must perform to ensure that your applications and operating environment will function properly. The specific tasks that are required depend on the specific OpenEdge components that you are upgrading.

The following steps describe the general workflow of upgrading to OpenEdge 12:

1. Prior to installing OpenEdge 12, Progress recommends that you review the following:

   - The deprecated and de-supported features and functionality that are listed in the *OpenEdge 12 Platform & Product Availability Guide*, available on https://docs.progress.com. This document also identifies the platform requirements for installing and running OpenEdge 12.

     See also "Before You Install" in *Install OpenEdge*.

   - *OpenEdge 12 Release Notes*.

2. Install the OpenEdge 12 software.
3. Complete the database upgrade steps described in Convert an OpenEdge Release 11 Database to OpenEdge Release 12 on page 11.

   Note: When your existing database is enabled for Transparent Database Encryption (TDE) and uses the RC4 encryption algorithm for any of its tables, indexes, LOBs, or areas, you must change the TDE policy for each RC4-encrypted object to use an encryption algorithm that is supported in OpenEdge 12. You must perform this update **before** you convert your database to version 12, as described in Convert an OpenEdge Release 11 Database to OpenEdge Release 12 on page 11.

4. Upgrade your application development environment, described in Development Environment Upgrade Tasks on page 23.
5. Optionally, create a new PAS for OpenEdge instance.

6. Rebuild all applications in your OpenEdge 12 development environment. When upgrading your ABL applications, note the tasks described in Upgrade ABL Applications on page 33.

Note: Progress recommends Progress Developer Studio for OpenEdge for use on development systems only. To upgrade your OpenEdge applications that are deployed in a production environment, first recompile and update them on a development system, then export the upgraded applications to the OpenEdge production system.

---

7. Update system configuration property settings as required, described in OpenEdge Management AdminServer Configuration Upgrade on page 17.
8. Upgrade OpenEdge security as described in Upgrade Security to OpenEdge 12 on page 29.
9. Be sure to read the information described in Migration notes on page 35.
10. When promoting your upgraded applications to testing, staging, and production environments, be sure to install OpenEdge 12 in those environments and create new PAS for OpenEdge instances that are based on OpenEdge 12.

2

Convert an OpenEdge Release 11 Database to OpenEdge Release 12

The PROUTIL CONV1112 utility converts the schema of an OpenEdge Release 11 database to OpenEdge Release 12.

Note: Converting without a replication rebase may allow a faster upgrade, for eligible versions of OpenEdge. See 'Prepare to convert replication-enabled databases' and 'Convert replication-enabled databases without target rebase' in *Upgrade to OpenEdge 12*.

To convert the schema of an OpenEdge Release 11 database:

1. Back up the database using OpenEdge Release 11.

   Caution: There is always a chance that your schema could become corrupt during conversion. If the conversion fails, your database cannot be recovered. If this happens, you must restore the backup copy of your database and begin the conversion again.

2. Verify that the database is not in use.
3. When your OpenEdge Release 11 database is enabled for Transparent Database Encryption (TDE) and has area, table, LOBs, or index data objects currently encrypted with RC4, you must complete the following steps before running the PROUTIL CONV1112 utility. If your database is not TDE-enabled, or is not using RC4 for any of its areas, tables, LOBs, or indexes, you may skip to step 4.
   a) Modify your OpenEdge Release 11 database to change the TDE policy for each RC4-encrypted table, index, LOB, or area to use an encryption algorithm that is supported in OpenEdge 12, such as AES-128.

For details, see “Change the cipher of an encryption policy with PROUTIL EPOLICY” in *Learn about Security and Auditing*.

b) Update the encrypted data that is stored in the database.

For an example, see “EPOLICY MANAGE UPDATE example” in *Learn about Security and Auditing* (an example of updating is described in steps 4 and 5).

c) Make sure that the updated tables, indexes, LOBs, and areas show that all of the stored data is encrypted with the new algorithm.

4. Truncate your before-image file with Release 11 PROUTIL TRUNCATE BI. You cannot truncate the before-image file of your Release 11 database with Release 12.
5. Disable after-imaging, if enabled, with Release 11 RFUTIL AIMAGE END.

   If you have OpenEdge Replication enabled, you must disable it before you can disable after-imaging.

   For information on Replication, see “OpenEdge Replication and After-imaging” in *Use Database Replication*.
6. Disable two-phase commit, if enabled, with Release 11 PROUTIL 2PHASE END.
7. Verify your Release 11 backup, and then install OpenEdge Release 12, if you have not already done so.

   For information about how to install OpenEdge, see “Before You Install” in *Install OpenEdge*.
8. Run PROUTIL CONV1112 to convert your database.

   Use the PROUTIL CONV1112 command, as shown:

```
proutil db-name -C conv1112
```

   CONV1112 performs extensive checks of your database during a pre-scan process. If the pre-scan encounters any issues that would prevent a successful conversion, the issues are logged and the convert utility exits without making any changes to the database (it is still a valid Release 11 database).
9. After you have successfully converted your database, back up your OpenEdge Release 12 database using Release 12 PROBKUP.

   - You should back up your Release 12 database in case it is damaged due to some failure. If you have only a Release 11 backup of your database, then you will need to go through the conversion process again.
   - Changes to large file support in OpenEdge Release 12 will impact databases created using CONV1112.

     For more information, see “Database creation and large file support” in *Manage the OpenEdge Database*.

For details, see the following topics:

- Prepare to convert replication-enabled databases
- Convert replication-enabled databases without target rebase

Prepare to convert replication-enabled databases

For conversion without a replication rebase, source and target databases must meet certain requirements.

The conversion utility CONV1112 performs a pre-scan to check for conditions that prevent a complete conversion. If any conditions exist that would prevent a complete conversion are found, they are reported, and CONV1112 exits normally with no change to the database. You must correct all reported issues before CONV1112 can successfully complete. Conditions that cause the pre-scan to exit include:

- The database is not an OpenEdge Release 11 database.
- The database has Transparent Data Encryption (TDE) enabled, and an encryption policy using the current or previous RC4 cipher (Cipher ID 7) exists on any table, index, or LOB.

Note: If existing encryption uses the RC4 cipher (Cipher ID 7), conversion is cancelled and you will receive a message that the cipher is no longer supported. You must then either:

- Use the PROUTIL EPOLICY utility to remove the RC4 cipher (Cipher ID 7) encryption from the encrypted objects, and PROUTIL EPOLICY UPDATE to retire the cipher; or
- Use the PROUTIL EPOLICY utility to change the RC4 cipher (Cipher ID 7) to a stronger encryption cipher, and use the PROUTIL EPOLICY UPDATE and PROUTIL EPOLICY SCAN operations to re-encrypt the objects with the new policy and cipher.

These operations may be performed on either the existing OpenEdge Release 11 database or on a backup created at the time of conversion.

To ensure that your source and target databases meet requirements for replication, follow these steps:

1. Back up your OpenEdge Release 11 source database using a Release 11 backup utility.
2. Verify that the source database is not in use, and that all transactions have committed or rolled back.
3. Make sure that replication is enabled on both source and target, and remains enabled on both source and target during conversion.
4. Make sure that after-imaging is enabled on the source, and remains enabled during conversion.
5. If after-imaging is enabled on the target(s), leave it enabled during conversion. If after-imaging is disabled on the target(s), leave it disabled during conversion.
6. Verify that the source and target databases are synchronized:
   a) Shut down both source and target(s) before the conversion.
   b) Restart source and target(s).
   c) Wait for synchronization to complete. Check the log files for the message that synchronization has completed.

      Restart the replication source database and look for the following message in the `sourcedb.lg` or the `sourcedb.repl.server.lg` file

      (10436) The source database [source] and the target database [target] on host [hostname] are synchronized.

      Restart the replication target database and look for the following message in the `targetdb.lg` or the `targetdb.repl.agent.lg` file

(10668) The Source and Target databases are synchronized.

d) Shut down both source and target(s) immediately after synchronization completes. Make sure that no new transactions start on the source, or the target can lose synchronization with the source again.

Caution: Do not allow new transactions after synchronization. Changes to the source database between synchronization/shutdown and conversion do not replicate to the target databases. The conversion of the target database will fail due to validation checks performed during conversion.

---

7. Truncate the BI file of the replication source with OpenEdge Release 11.

   For instructions, see 'PROUTIL TRUNCATE BI qualifier' in *Manage the OpenEdge Database*.

Now you are ready to convert your replication-enabled source and target databases, as described in Convert replication-enabled databases without target rebase on page 14.

Convert replication-enabled databases without target rebase

The conv1112 utility lets you convert replication-enabled databases from OpenEdge release 11.x to OpenEdge release 12.3 without requiring a rebase of the target database(s).

Caution: Before you begin, make sure that you complete the steps in Prepare to convert replication-enabled databases on page 13.

By avoiding a time-consuming rebase, a database schema conversion is potentially faster than a full database conversion. You can convert without rebase when the only database changes from OpenEdge Release 11 are in the schema area. OpenEdge Release 12.3 is the only release that currently supports conversion without rebase. For example, you can run a schema conversion to convert a replication-enabled database from Release 11.x to Release 12.3.

To convert the schema of a replication target database:

1. Run the conv1112 utility offline with `replsource` on the OpenEdge Release 11.x source database. For example, to create a backup file named `sports2020schemabk` for the replication-enabled `sports2020` database::

   ```
   proutil sports2020 -C conv1112 replsource sports2020schemabk
   ```

   For details, see 'PROUTIL CONV1112 REPLSOURCE qualifier' in *Manage the OpenEdge Database*.

2. Next, run the conv1112 utility offline with `repltarget` on the OpenEdge Release 11.x target database.

   ```
   proutil sports2020 -C conv1112 repltarget sports2020target
   ```

   For details, see 'PROUTIL CONV1112 REPLTARGET qualifier' in *Manage the OpenEdge Database*.

   During the conversion:

   - The generated replication source has after-imaging and replication enabled.
   - AI extents existing on disk prior to conversion are converted and usable with the OpenEdge replication source after conversion.

- If your encryption is enabled on your database, it is temporarily disabled, preventing access to the encryption cache and any encrypted data in the database. Conversion completion re-enables encryption and restores access to the encryption cache and any encrypted data.
- If your database has Change Data Capture (CDC) enabled, it is temporarily disabled, preventing CDC cache allocation and loading. CDC is re-enabled when the conversion completes.
- All schemas in your database are updated to the latest version, and any new schema tables are added.
- All existing 32-bit sequences are upgraded to 64-bit.

  Existing 32-bit sequences set to cycle at the limit, with the unknown value (?) as the limit, are changed to cycle at the maximum or minimum value of a 32-bit integer (+/-2147483647), keeping the behavior of a cycling sequence consistent.

  Existing 32-bit sequences that do *not* have a cycle and the limit is the unknown value (?), are changed to have an implied maximum or minimum of a 64-bit integer (+/-9223372036854775807).

  Caution: Verify any impact the sequence conversion may have on your application. Review your database sequence schema after the conversion completes, to verify initial and cycle settings, or take advantage of the 64-bit values.

- All virtual system tables (VSTs) in your database are recreated and updated to the latest version, and any new VSTs are added.
- New auditing events are populated.

During the conversion:

- The generated replication source has after-imaging and replication enabled.
- AI extents existing on disk prior to conversion are converted and usable with the OpenEdge replication source after conversion.
- If your encryption is enabled on your database, it is temporarily disabled, preventing access to the encryption cache and any encrypted data in the database. Conversion completion re-enables encryption and restores access to the encryption cache and any encrypted data.
- If your database has Change Data Capture (CDC) enabled, it is temporarily disabled, preventing CDC cache allocation and loading. CDC is re-enabled when the conversion completes.
- All schemas in your database are updated to the latest version, and any new schema tables are added.
- All existing 32-bit sequences are upgraded to 64-bit.

  Existing 32-bit sequences set to cycle at the limit, with the unknown value (?) as the limit, are changed to cycle at the maximum or minimum value of a 32-bit integer (+/-2147483647), keeping the behavior of a cycling sequence consistent.

  Existing 32-bit sequences that do *not* have a cycle and the limit is the unknown value (?), are changed to have an implied maximum or minimum of a 64-bit integer (+/-9223372036854775807).

  Caution: Verify any impact the sequence conversion may have on your application. Review your database sequence schema after the conversion completes, to verify initial and cycle settings, or take advantage of the 64-bit values.

- All virtual system tables (VSTs) in your database are recreated and updated to the latest version, and any new VSTs are added.

- New auditing events are populated.

After conversion, source and target databases are converted to OpenEdge Release 12.3.

3

OpenEdge Management AdminServer Configuration Upgrade

In releases of OpenEdge prior to version 12, the OpenEdge Management web application was hosted on Eclipse Jetty. Eclipse Jetty was, in turn, hosted as a plugin to the AdminServer. In OpenEdge 12, OpenEdge Management is hosted on a dedicated instance of Progress Application Server (PAS) for OpenEdge. This allows OpenEdge Management to run independently of the AdminServer, and also to manage multiple local as well as remote AdminServer instances.

Each PAS for OpenEdge instance has a `conf/appserver.properties` file in which you make updates to system properties. Several of the properties that were previously defined in the `fathom.init.params` file are moved into this `conf/appserver.properties` file as-is. Note the following updates that are required to the AdminServer configuration of OpenEdge Management.

System properties configuration file

Some properties in the `appserver.properties` file are new in OpenEdge 12 and replace values that were previously stored in the `fathom.init.params` file, as follows:

```
#Fri, 11 Jan 2019 18:03:26 +0530
# The appserver.properties file contains name-value Java properites used by
# the Progress Application Server (PAS) and web applications.
# Each non blank and non comment line will be defined as a java system property
# by prepending a '-D' and passed on the command line to the JVM via the
# JAVA_OPTS environment variable.
#
# Comment lines may be used by placing a pound signe (#) in column zero.  Trailing
# comments are not supported.
# You may use shell environment variable substitution using the form ${env-var-name}

# The release version of the Progress Application Server version
psc.as.version=4.0.0

# The type of PAS installation {standard|instance}
psc.as.type=instance

# The PAS installation OS type {windows|unix}
psc.as.os=windows

# The PAS installation personality type {appserver|tokenserver|fathom}
psc.as.personality=fathom


# windows service name used in windows registry for the service name
psc.as.winsvcname=cat


# Optional PAS instance name for tcman admin tools.  If not specified
# the instance name is the CATALINA_BASE's directory name "home"
psc.as.alias=ho

# Optional directory path of a PAS instance's parent
psc.as.parent=C:/oemgmt

# The PAS server & instances security policy {default|developer|production}
psc.as.security.model=production

# Indicator {true|false} of Windows OS service registration
psc.as.service=true

#OpenEdge installation directory needed for classic Adminserver libraries
Install.Dir=C:/dlc

#OpenEdge installation directory needed for PAS libraries
psc.as.oe.dlc=C:/dlc

#Fathom configuration file location
psc.oem.fathom.properties=${CATALINA_BASE}/conf/fathom.properties

#Fathom log file directory
psc.oem.logs=${CATALINA_BASE}/logs

#Fathom SNMP adapter state directory
psc.oem.snmp=${CATALINA_BASE}/work/snmp

#Fathom installation directory
psc.oem.install=${CATALINA_BASE}

#Fathom log file
psc.log.default=${CATALINA_BASE}/logs/fathom.log

#Fathom exception log file
psc.log.exceptions=${CATALINA_BASE}/logs/exceptions.log

#logging level for Fathom messages
LogLevel=3

#Local directory for Fathom configuration database
```

```
# configuration database will be created in a subdirectory of the specified directory
psc.oem.configdb.dir=${CATALINA_BASE}/work/

#Local directory for Fathom cache database
# graph cache database will be created in a subdirectory of the specified directory
psc.oem.activitydb.dir=${CATALINA_BASE}/work/

#User file location for Fathom logins
psc.oem.realm.properties=${CATALINA_BASE}/conf/fathomRealm.properties

#Permission policy file for Fathom user roles
psc.oem.realm.policy=${CATALINA_BASE}/conf/fathomRealm.policy

#Default location for Fathom trend database
psc.oem.fathom.trend=${CATALINA_BASE}/work/fathomdb/fathom.db

#OrientDB tuning parameters for cache and configuration database
storage.record.lockTimeout=75000

# max segment size for write ahead log (BI file equivalent) for cache and configuration
 databases
storage.wal.maxSegmentSize=32

# max size in MB of write ahead log for cache and configuration databases (value applies
 separately to each)
storage.wal.maxSize=256

# amount of memory in MB to allocate for cache and configuration databases
storage.diskCache.bufferSize=384

# controls memory allocation and pooling for lock manager pool
environment.lockManager.concurrency.level=8

# controls the number of files created per cluster for cache and configuration databases
class.minimumClusters=1

# Disable directIO access.  OrientDB use of Direct IO on Linux fails if the file system
 is on an NFS mount. This disables
# direct IO which is a bit sloewr, but prevents the file open problems.  This can safely
 be enabled
# on Windows, and non-NFS mounted drives on *nix.
storage.wal.allowDirectIO=false

#OpenEdge Management report generation directory. If not provided this uses
$OEMWRKDIR/reports directory
#You may override the report directory by specifying an alternate value here
#psc.oem.reports=

# ensure snmp4j sends logging information to log4j
snmp4j.LogFactory=org.snmp4j.log.Log4jLogFactory

# Configure SLF4J compatible logger for OEM
psc.log.factory=com.progress.fathom.logging.LogSystemBridgeToSlf4JFactory

# Logback configuration file
logback.configurationFile=${CATALINA_BASE}/conf/logging.xml
```

fathom.properties

In OpenEdge 12, the `fathom.properties` file is moved from the `<DLC>/properties` directory to the `<OEMgmt>/conf` directory. In addition, some of the properties defined in this file are not used in Release 12. Some properties may still be viewable, but you must update the location in which they are stored if you want to retain them.

The following table lists the properties that are affected by the transition to Tomcat. Any properties not listed in this table remain unchanged in Release 12.

Table 1: Property changes

Property Name	Description
WebServer.HttpEnabled	This property remains in the user interface, and you can set it by using TCMAN to enable and disable the HTTP connector.
WebServer.HttpPort	This property remains in the user interface and is stored in the `<instance>/conf/catalina.properties` file as the*psc.as.http.port* property.
WebServer.HttpsEnabled	This property remains in the user interface, and you can set it using TCMAN to enable and disable the HTTP connector.
WebServer.HttpsPort	This property remains in the user interface and is stored in the `<instance>/conf/catalina.properties` file as the *psc.as.https.port* property.
SSLIdentityAlias	This property remains in the user interface and is stored in the `<instance>/conf/catalina.properties` file as the *psc.as.https.keyalias* property.
SSLIdentityAliasPassphrase	This property remains in the user interface and is stored in the `<instance>/conf/catalina.properties` file as the *psc.as.https.keypass* property.
SSLIdentityKeystorePathname	

4

Development Environment Upgrade Tasks

The tasks associated with upgrading your development environment to the current release of OpenEdge include:

- Upgrading the Spring Security files that are contained in ABL WebApp projects that you have imported into the current version of Progress Developer Studio. Note also that the means by which domains are used to seal and validate client-principal objects has also been updated in OpenEdge 12.
- Updating the `build.xml` file that is used to package an ABL web application ANT project.
- Upgrading your ABL Service projects that are created in Progress Developer Studio, and creating a new PAS for OpenEdge Release 12 instance.

For details, see the following topics:

- Upgrade an ABL WebApp project
- Package an ABL web application ANT project
- Upgrade ABL Service Projects in a PAS for OpenEdge Instance

Upgrade an ABL WebApp project

If you have imported an ABL WebApp project from an older OpenEdge version to the current version of Progress Developer Studio, you must upgrade the **Spring Security** files.

To upgrade the Spring Security files, complete the following steps:

1. Right-click the ABL WebApp project and select **Progress OpenEdge** > **Migrate Spring Security Files**. The **Migrate Spring Security Files** dialog box also opens.

   Note: The dialog box also opens when you open an old workspace or import an old project to the lastest workspace that contains ABL WebApp projects.

2. Click **OK**.

   Note: If you click **Cancel**, you must later follow the previous step to upgrade.

   The Spring Security files in the current `WEB-INF` folder in the project directory are moved to the newly created backup folder. The migrated Spring Security files are placed in the `WEB-INF` folder.

Updates to the cryptography used to secure client-principal objects

The means by which domains are used to seal and validate client-principal objects has been updated in OpenEdge 12. As a result, any static client-principal objects that were sealed by an ABL application in a version of OpenEdge prior to Release 12 cannot be validated in OpenEdge 12. Any client principal used in OpenEdge 12 can be used for identity or validated as your identity in OpenEdge 12 only if it was sealed by an OpenEdge 12 domain.

Package an ABL web application ANT project

To package an ABL web application ANT project in release 12, you need to update the existing `build.xml` file. Add the following code in `build.xml`:

```
<pathelement location="${dlc.java.dir}/ext/jettison-1.2.jar"/>
<pathelement location="${dlc.java.dir}/ext/commons-logging-1.2.jar"/>
<pathelement location="${dlc.java.dir}/ext/xmlschema-core-2.2.1.jar"/>
```

For more details, see “Package an ABL WebApp ANT project” in *Progress Developer Studio for OpenEdge Online Help*.

Upgrade ABL Service Projects in a PAS for OpenEdge Instance

Before you upgrade a PAS for OpenEdge instance and its deployed applications from OpenEdge Release 11.7 to 12, be sure to create a backup your existing PAS for OpenEdge instance and its deployed instances. You can do this by creating a zip archive of the root directory of the OpenEdge 11.7 instance. And if you are not using a source code control system, then Progress also recommends that you also create a backup of the entire project space.

After your backup is complete, upgrade all of your ABL Service projects that are created in Progress Developer Studio, and then create a new PAS for OpenEdge Release 12 instance. For details, complete the steps described in the following topics:

Upgrade ABL Service projects

To upgrade an ABL Service project to OpenEdge Release 12, you have two options:

- Create a new project in OpenEdge 12, and then copy in the source files and support files from the OpenEdge 11-based workspace.
- Manually upgrade the existing OpenEdge 11-based project.

If you choose the latter option, then complete the following steps for each project. Performing these steps ensures that all configuration properties are fully merged into your OpenEdge 12 development environment:

1. Copy the `PASOEContent/WEB-INF/logging.xml` file into the `PASOEContent/WEB-INF/backup` folder.
2. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/logging.xml` file into `PASOEContent/WEB-INF` folder. Note that this step overwrites the older `logging.xml` file.
3. Copy the `PASOEContent/WEB-INF/oeablSecurity.properties` file into the `PASOEContent/WEB-INF/backup` folder.
4. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/oeablSecurity.properties` file into the `PASOEContent/WEB-INF` folder.
5. Merge any changes from the `PASOEContent/WEB-INF/backup/oeablSecurity.properties` file into the `PASOEContent/WEB-INF/oeablSecurity.properties` file.
6. Copy the `PASOEContent/WEB-INF/oauth2ResSvcClients.cfg` file into the `PASOEContent/WEB-INF/backup` folder.
7. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/oauth2ResSvcClients.cfg` file into `PASOEContent/WEB-INF` folder. Note that this step overwrites the older `oauth2ResSvcClients.cfg` file.
8. Merge any changes from the `PASOEContent/WEB-INF/backup/oeablSecurity.properties` file into the `PASOEContent/WEB-INF/oauth2ResSvcClients.cfg` file.
9. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/oeablSecurity.xml` file into the `PASOEContent/WEB-INF` folder.
10. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/spring` folder and its contents into the `PASOEContent/WEB-INF/spring` folder.
11. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/web.xml` file into the `PASOEContent/WEB-INF` folder.
12. Copy the *`%dlc%`*`/servlets/oeabl/WEB-INF/home` folder and its contents into the `PASOEContent/WEB-INF` folder.
13. Copy the *`%dlc%`*`/servlets/oeabl/static/common*` files into the `PASOEContent/static` folder.
14. Copy the contents of the *`%dlc%`*`/servlets/oeabl/static/images` folder into the `PASOEContent/static/images` folder.
15. Create new PAS for OpenEdge instances that are based on OpenEdge Release 12, as described in Create the PAS for OpenEdge instance on page 26.
16. Deploy the ABL Service projects as PAS for OpenEdge web applications in the newly-created PAS for OpenEdge instances.

Create the PAS for OpenEdge instance

The best practice for upgrading a PAS for OpenEdge instance is to create a new instance that is based on OpenEdge Release 12, and to merge changed files into that new instance. You then take the old instance out of service. Note that if the new instance is configured with the same port number as the old instance, then you must stop the old instance before you are able to start the new one.

To upgrade a PAS for OpenEdge instance, complete the following steps:

1. Create a new PAS for OpenEdge instance by entering the following command:

   ```
   pasman create
   ```

   This command creates a default ABL application that uses the ROOT web application.

2. Deploy upgraded web applications by entering the following command. If you are not using the PAS for OpenEdge instance name as the default ABL application name, then make sure that you specify the ABL application name correctly.

   ```
   tcman deploy
   ```

3. Copy files from the old PAS for OpenEdge instance to the new instance, as follows:

   a. Copy any files and folders in all of the `ablapps/`*`abl-app-name`* directories, excluding the `conf/oeablSecurity.properties` file.

   b. Merge properties that are specified in the *`/path-to-old-instance/`*`ablapps/`*`abl-app-name`*`/conf/oeablSecurity.properties` file into the *`/path-to-new-instance/`*`ablapps/`*`abl-app-name`*`/conf/oeablSecurity.properties` file.

   c. Copy ABL application lifecycle scripts from the *`/path-to-old-instance/`*`bin` directory to the *`/path-to-the-new-instance/`*`bin` directory. This includes the following:

      - *`abl-app-name`*`_setup.{bat|sh}`
      - *`abl-app-name`*`_startup.{bat|sh}`
      - *`abl-app-name`*`_started.{bat|sh}`
      - *`abl-app-name`*`_shutdown.{bat|sh}`

   d. Copy product OS environment variable setup scripts from the *`/path-to-old-instance/`*`bin` directory to the *`/path-to-the-new-instance/`*`bin` directory. This includes the *`your-name`*`_setenv.{bat|sh}` script.

   e. Copy or merge any other product shell scripts in the *`/path-to-old-instance/`*`bin` directory to the *`/path-to-new-instance/`*`bin` directory.

   f. Copy any source or rcode file from the *`/path-to-old-instance`* directory to the *`/path-to-new-instance`* directory.

   g. Copy any other installation- or application-specific folders from the *`/path-to-old-instance`* directory to the *`/path-to-new-instance`* directory.

4. Merge properties from the older instance into the new, as follows:

   a. For OpenEdge-specific properties, use the following command:

```
/path-to-new-instance/bin/oeprop -f
/path-to-old-instance/conf/openedge.properties
```

b. Merge any other properties file using a text merge tool.

5. Merge PAS for OpenEdge features from the old instance to the new, as follows:

a. Obtain the feature configuration by entering the following command:

```
/path-to-old-instance/bin/tcman.{bat|sh} feature
```

b. Make any changes by entering the following command:

```
/path-to-new-instance/bin/tcman.{bat|sh} feature feature-name={on|off}
```

6. Make sure the ROOT web application is not in use, and then remove it by entering the following command:

```
tcman undeploy ROOT
```

A best practice is to replace the ROOT web application with an empty or blank Java web application that satisfies Tomcat's default web application context lookup.

7. Test the PAS for OpenEdge instance before adding it to your production environment.

5

Upgrade Security to OpenEdge 12

The following topics describe important notes regarding security updates made in OpenEdge Release 12, and also describe tasks you may need to complete to upgrade your application environment security to Release 12.

For details, see the following topics:

- Updates to password encryption with GENSPRINGPWD
- PAS for OpenEdge and OpenEdge Authentication Gateway: Domain configuration
- Secure communications in OpenEdge Management
- Upgrade an OpenEdge Authentication Gateway instance configuration

Updates to password encryption with GENSPRINGPWD

PAS for OpenEdge supports the use of the extended local user accounts model, which allows you to store encrypted user passwords in the `users.properties` file that is local to your instance's web application. OpenEdge 11.7 supports several hashing algorithms for password encryption: `bcrypt`, `sha256`, `sha512`, and `sha1`. However, in OpenEdge 12, the only supported algorithm is `bcrypt`, which is set by default. The other algorithms are not supported in OpenEdge 12.

If you encrypted passwords with `sha256`, `sha512`, or `sha1` in your OpenEdge 11.7 applications, then complete the following steps to upgrade the hashing algorithm to `bcrypt`:

1. Obtain the original clear-text (unencrypted) password that you encrypted with `GENSPRINGPWD`.
2. Encrypt the password with `GENSPRINGPWD` in OpenEdge 12. This generates the encrypted password using `bcrypt`.
3. Copy the encrypted password and insert it into the corresponding entry in the instance's `users.properties` file.

The following is the syntax for storing passwords in the `users.properties` file:

```
username=password, role1, [role2, ..., roleN, ] { enabled | disabled }
```

For more information about setting the extended local user accounts model, see "About Local and Extended Local user accounts" in *Manage Progress Application Server (PAS) for OpenEdge*.

PAS for OpenEdge and OpenEdge Authentication Gateway: Domain configuration

In OpenEdge, the authentication system in use is identified by an OpenEdge security domain, or simply **domain**. To authenticate a user identity, an authentication system must:

- Be enabled for user authentication
- Have access to a valid set of user accounts
- Be part of an OpenEdge domain configuration that is enabled for run time

This configuration is necessary for a domain to be enabled for authentication. An identity that is being authenticated must provide a user name and a domain name so that the fully-qualified `user ID` can be established.

Domain access codes are used by the PAS for OpenEdge and Authentication Gateway servers to create client-principal objects. In OpenEdge 12, the ability to configure domain access codes in the PAS for OpenEdge and Authentication Gateway configuration properties file is no longer supported. Instead, you must explicitly define domain access codes in a CSV file, which is encrypted into the domain registry keystore, `domains.keystore`.

For more information, see "Configuring domain access codes" in *OpenEdge Getting Started: OpenEdge Authentication Gateway Guide*. If you use the properties file to configure the domain access code, the server instances will not start.

Explictly defining domains and their access codes is the default functionality in Release 12 of OpenEdge Authentication Gateway. Consequently, the domain access code 'key' property is removed from the `sts.properties` file.

Note that for backward compatibility, the PAS for OpenEdge configuration includes a built-in OpenEdge domain registry keystore that supplies the same default blank domain and its blank access code as provided in earlier releases. If your 11.7.x PAS for OpenEdge application security configuration:

- Uses only the default settings, then no upgrade is required.
- Uses a domain registry keystore (with a list of domains and their access codes defined), then generate the registry keystore again.

  To generate the keystore, use the `'gendomreg'` utility, in *Manage OpenEdge Keys and Certificates*.
- Allows multiple domains, but uses the default blank domain access code to seal client-principal objects, then you must remove the 'key' property from the `sts.properties` file and use the OpenEdge domain registry keystore file.

Secure communications in OpenEdge Management

In OpenEdge 12, both HTTP and HTTPS transport ports are enabled by default in each instance. If you upgrade your resource configuration from OpenEdge 11.7.x to OpenEdge 12, and you had the HTTPS port disabled in your 11.7.x version, the HTTPS port will be enabled automatically in OpenEdge 12.

To disable secure communications (HTTPS) in your OpenEdge 12 environment, perform either of the following actions:

1. Manually edit the `catalina.properties` file in the `$OEMGMT/conf` directory to comment out the `psc.as.https.port` property.
2. Use the OpenEdge Management console to disable the HTTPS port.

Upgrade an OpenEdge Authentication Gateway instance configuration

OpenEdge Authentication Gateway instances created in OpenEdge 11.6.x or 11.7.x are not compatible with OpenEdge 12.

To upgrade your Authentication Gateway instance configuration from 11.7.x to Release 12, complete the following steps:

1. Back up the instance to save your 11.7.x configuration.
2. Create a new Authentication Gateway server instance during the OpenEdge 12 installation process.
3. Upgrade the 11.7.x configuration from the backup to the new Version 12 instance:
   a) Configure the HTTPS port (the procedure is unchanged from 11.7.x).
   b) Copy the `tomcat-keystore.p12` file, which has the server instance's TLS private key and server certificate, to the Version 12 instance (`/oeauthserver/conf`).
   c) Merge any HTTPS keystore and password configuration.
   d) Merge any configuration changes made to the `conf/openedge.properties` file from the 11.7.x instance using your preferred merge tool.
   e) Copy the `tomcat-users.xml` file from the 11.7.x instance to the Version 12 instance (`/oeauthserver/conf`).

f) Copy any `.r` ABL application files from the 11.7.x instance (OERealm, Policy, and Event handlers) to the Version 12 instance.

g) Copy the OpenEdge domains configuration file (`domains.json`) from the 11.7.x instance to the Version 12 instance (`/oeauthserver/webapps/ROOT/WEB-INF/config` directory).

h) Copy the `users.properties` file from the 11.7.x instance to the Version 12 instance (`/oeauthserver/webapps/ROOT/WEB-INF/config` directory).

i) Create a new domain registry keystore file (`domains.keystore`) from the 11.7.x instance's `domainsks.csv` file using the `gendomreg` utility, and add the new `domains.keystore` file to the `/oeauthserver/webapps/ROOT/WEB-INF/config` directory in the Version 12 instance.

j) Merge the 11.7.x instance `sts.properties` configuration file with the Version 12 instance `sts.properties` file using your preferred merge tool.

4. If you have Security Token Service (STS) key security enabled for your 11.7.x instance, then:

   a) Create a new Version 12 STS server key file using the same `-url` parameter as was used to generate the 11.7.x server-key file. Store the new STS server key file in the location that is specified as the value of the `sts.server.keyfile` property in the `sts.properties` file.

   b) Install a new STS client key file on the local server using the `stskeyutil` tool.

   c) Test the STS server key access using the `stskeyutil` tool.

5. Start the Version 12 instance, and then test access to the instance OpenEdge domain configurations using the `stsclientutil` tool.

6. If you have enabled STS key security for your Authentication Gateway, then:

   a) Install a new STS client key in all OpenEdge 12 client installations that replaced the 11.7.x installations.

   b) Test the client key access using the `stsclientutil` tool.

7. Test the Version 12 PAS for OpenEdge and OpenEdge database connections that are configured to use the Authentication Gateway.

6

Upgrade ABL Applications

OpenEdge Release 12 introduces some security changes that may require you to upgrade your ABL applications, depending on the specific OpenEdge features that your ABL applications use.

ABL callbacks

If the OpenEdge security domain configuration in your OpenEdge Database employs ABL callbacks, you must update the domain configuration by re-registering the new version 12 compiled module. Use the database Data Administration tool to register the version 12 module in the Authentication System dialog box.

Security domain and system tables loaded from dump file

If your product's database installation loads the OpenEdge domain and system tables from a dump (`.d`) file, you will need to migrate your dump file for use in creating a version 12 database. To perform this migration, you must define your product's default OpenEdge domain and system tables in a version 12 database, and then use the Data Administration utility to create a new dump file.

Additional changes that may be needed to application code

You might also need to update your ABL application code in the following cases:

- If you use an `oech1::` encoded OpenEdge domain access-code in the `CLIENT-PRINCIPAL:SEAL()` method, you must update the code to pass a clear-text value as the method parameter.
- If you use the `AUDIT-POLICY:ENCRYPT-AUDIT-MAC-KEY()` method to manage OpenEdge domain access codes in your database, you must replace this method with the `SECURITY-POLICY:ENCODE-DOMAIN-ACCESS-CODE()` method.
- If you use the `RCODE-INFO:MD5-VALUE` attribute, you must replace it with the `RCODE-INFO:SIGNATURE-VALUE` attribute. **Note** that the size of the return value increases from 16 bytes of storage to 32 bytes.

- If you use object serialization between ABL clients and application servers, you must migrate both clients and server to version 12.
- If you use object serialization to persistent storage (for example, file system or database), you must recreate the object data in the persistent storage. Serialized data written in a version 11 ABL application is incompatible with a version 12 ABL client's ability to read that data.

7

Migration notes

When upgrading from OpenEdge 11.7.x to OpenEdge 12, bear in mind the following:

- If you import OpenEdge 11.7.x installation configurations using the `configutil` utility, then the property files associated with the classic AppServer and the configurations related to OpenEdge Management, OpenEdge Business Process Management, and Tomcat-in-the-box will be ignored.

  For more information about the `configutil` utility, see “Manage Installation Configurations” in *Install OpenEdge*.

- If you migrate your resource configuration from OpenEdge Management 11.7.x to OpenEdge Management 12 , then the following in the fathom.xml file will be ignored during the import process:

  - AppServer
  - AppServer Internet Adapter (AIA)
  - ODBC DataServer
  - REST resources
  - SonicMQ Adapter
  - WebSpeed
  - Web Services Adapter (WSA)