Progress®



OpenEdge® Service Pack 11.7.5:
New Information

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.

May 2019

Last updated with new content: Release 11.7.5

Updated: 2021/06/28

Table of Contents

Preface....7
Purpose....7
Audience....7

What's New in 11.7.5....9
OpenEdge SQL....9
Troubleshoot a SQL crash....10
REVOKE statement for sequences documentation for the SQL Reference....11
OpenEdge ABL....13
STOP condition handling using stop objects is fully supported....13
OpenEdge Management....14
Set OpenEdge Replication Server properties....14
Set OpenEdge Replication Agent properties....15
Set OpenEdge Replication Transition properties....17
Configure parameters and limits for Progress Application Server for OpenEdge....20
Install other web applications shipped with Progress Application Server for OpenEdge....22
Configure database environment variables....22
PAS for OpenEdge....23
Manage security properties using SECPROP....23
Change to metrics returned by getMetrics....28
Progress Developer Studio for OpenEdge....30
Security updates....30

Preface

For details, see the following topics:

- Purpose
- Audience

Purpose

OpenEdge® Service Pack 11.7.5: New Information documents important information about the OpenEdge Release 11.7.5 Service Pack, with reference to new and existing documentation for Release 11.7 as required. Release 11.7 documentation set references might include:

- Product manuals
- HTML-based online help
- Web papers

References to these documents attempt to identify sections or topics that you can use in searches of the product documentation set.

Audience

This guide is primarily intended for OpenEdge application developers and system administrators who have installed 11.7.5 as the most recent Service Pack for Release 11.7.

1

What's New in 11.7.5

This topic describes the new features included in OpenEdge Release 11.7.5.

For details, see the following topics:

- OpenEdge SQL
- OpenEdge ABL
- OpenEdge Management
- PAS for OpenEdge
- Progress Developer Studio for OpenEdge
- Security updates

OpenEdge SQL

The OpenEdge Release 11.7.5 Service Pack contains changes to the SQL features described in the following sections.

Troubleshoot a SQL crash

When a SQL process terminates abnormally, it automatically creates a text file named `protrace.`*`pid`* in the current directory. In prior releases, the `protrace.`*`pid`* file contains only 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.

To help identify and diagnose the root cause of a SQL crash, the relevant SQL query is also now captured in the `protrace.`*`pid`* file, along with additional information required for debugging such as user name, database name and so on.

Note: Currently, OpenEdge support for capturing the SQL statement in the `protrace.`*`pid`* is available on Linux platforms only.

Example

```
command line not initialized
#1   [0xa43540] dbut_uttraceback+0x15f  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#2   [0xa43139] dbut_uttrace+0x1f4  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#3   [0xa16d64] dbExit+0x42  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#4   [0x9c438d] drSigFatal+0x11f  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#5   [0xd58326a0] __rpc_thread_destroy+0xffef1230  from /lib64/libc.so.6
#6   [0xd600f6ab] _fini+0xffffe4bb  from /lib64/libpthread.so.0
#7   [0x812a07] _ZL14dt_fpe_handlerv+0x57  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#8   [0xd58326a0] __rpc_thread_destroy+0xffef1230  from /lib64/libc.so.6
#9   [0x4986d9] _ZN9rds_env_t12crash_serverE11sqlm_type_t+0xa3  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#10  [0x4f3ce3]
_Z8ssm_openP9rds_env_tPKcP12sql_stmuid_thS2_S2_P5sqldaP5sqlcaS2_+0x1122  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#11  [0x4bfc73] sql_fn+0x9c7  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#12  [0x90ec62] _ZN14cursor_reply_t15process_requestEv+0x306  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#13  [0x90dc00] _ZN14server_reply_t13execute_replyEPPhPi+0x10a  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#14  [0x90e942] _ZN14cursor_reply_t7executeEPhPS0_PiS2_11sqlm_type_t+0x72  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#15  [0x90cb50] _ZN12srv_nw_hdl_t13dsn_listen_10EiPvP10svr_args_t+0x890  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#16  [0x7dd503] dmn_main+0x121  from /vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#17  [0x430dad] _Z14run_sql_threadPv+0x285  from
/vobs_sql/sql/linuxx86_64/bin/_sqlsrv2
#18  [0xd6007aa1] _fini+0xffff68b1  from /lib64/libpthread.so.0

 Following SQL Query resulted in the crash:
   select count(*) from pub.customer

 Additional information:
   User Name: hum
   DB   Name: testdb
   Schema  Name: HUM
   Catalog Name: TESTDB
   Audit Status : DISABLED
   Authorised Data Truncation Status: OFF
```

REVOKE statement for sequences documentation for the SQL Reference

The REVOKE statement revokes various privileges from the specified users of the database. There are two forms of the REVOKE statement:

- Revoke database-wide privileges, either system administration (DBA), general creation (RESOURCE), audit administration (AUDIT_ADMIN), audit archive (AUDIT_ARCHIVE), or audit insert (AUDIT_INSERT)
- Revoke various privileges on specific tables and views, sequences, stored procedures and user defined functions

Note: The REVOKE statement for sequences and stored procedures has been newly included.

Syntax

```
REVOKE { RESOURCE, DBA, AUDIT_ADMIN, AUDIT_ARCHIVE, AUDIT_INSERT}
   FROM {user_identifier[ , user_identifier] , ...}
   [ RESTRICT | CASCADE ]
   [ GRANTED BY ANY_USER ];
```

Parameters

RESOURCE

Revokes the privilege to issue CREATE statement for specified users.

DBA

Revokes the privilege to create, access, modify, or delete any database object, and revokes the privilege to grant other users any privileges for specified users.

AUDIT_ADMIN

Revokes the privilege to administer and maintain a database auditing system for specified users.

AUDIT_ARCHIVE

Revokes the privilege to read and delete audit records for specific users.

AUDIT_INSERT

Revokes the privilege to insert application audit records for specified users.

FROM *user_identifier*[, *user_identifier*] , ...

Revokes the specified privileges on the table or view from the specified list of users.

EXECUTE

Revokes the EXECUTE privileges for the specified list of users from using user defined functions and stored procedures.

RESTRICT | CASCADE

Prompts SQL to check whether the privilege being revoked was passed on to other users. If the original privilege was included the WITH GRANT OPTION clause, then the REVOKE statement fails and generates an error. The REVOKE statement succeeds if the privilege was not passed on.

If the REVOKE statement specifies CASCADE, then revoking the access privileges from a user also revokes the privileges from all users who received the privilege from that user.

If the REVOKE statement does not specify either RESTRICT or CASCADE, then the behavior follows only CASCADE.

user_identifier

Identifies a username. For a tenant user, you must mention the fully qualified user name, username@domain_name, to revoke access from a user.

Uses the following syntax:

```
username | username@domain_name
```

Note: CASCADE is not supported for AUDIT_ADMIN, AUDIT_ARCHIVE, and AUDIT_INSERT privileges. The only user who can revoke an audit privilege is the user who granted it.

GRANTED BY ANY_USER

A database administrator can use this phrase to revoke all access privileges to a table for a specified user, even if the user was the creator of the table. This phrase is only applicable to users with database administrator privileges.

Syntax to revoke privileges on stored procedures

Use the following syntax to revoke privileges for a user, or users, on specified procedures:

```
REVOKE EXECUTE ON StoredJavaProcedureName
 FROM  {username[, username] , . . .|  PUBLIC }
   [ RESTRICT | CASCADE ];
```

Syntax to revoke sequence privileges

Use the following syntax to revoke sequence privileges:

```
REVOKE [SELECT | UPDATE]
ON SEQUENCE schema.sequence
FROM user_name[,user_name]...
```

Parameters

SELECT

For the specified users, removes the privilege to read data from the sequence.

UPDATE

For the specified users, removes the privilege to modify data for the sequence.

Examples

In this example, the audit administration privilege is revoked from bsmith:

```
REVOKE AUDIT_ADMIN FROM bsmith RESTRICT;
```

Revoking privileges to modify a sequence

In this example, the UPDATE permission of the sequence pub.customer_sequence is revoked from the user slsadmin:

```
REVOKE UPDATE
     ON SEQUENCE pub.customer_sequence     FROM slsadmin;
```

OpenEdge ABL

OpenEdge Release 11.7.5 Service Pack adds support for the ABL features described in the section that follows.

STOP condition handling using stop objects is fully supported

STOP condition handling using stop objects was a technology preview feature in OpenEdge release 11.7.0. It is now a fully implemented and supported feature for OpenEdge ABL as of 11.7.3. However, you must still opt-in to the new behavior by using -catchStop 1. (The new behavior is the default in 12.0.)

Background

In 11.7.0 five new built-in objects were introduced:

- Progress.Lang.StopError
- Progress.Lang.Stop
- Progress.Lang.StopAfter
- Progress.Lang.UserInterrupt
- Progress.Lang.LockConflict

In 11.7.3, functionality was introduced to allow FINALLY blocks to execute as part of stop processing.

OpenEdge Management

OpenEdge Release 11.7.5 Service Pack adds support for the OpenEdge Management feature enhancements described in the following sections.

Set OpenEdge Replication Server properties

You set the OpenEdge Replication Server properties to specify server and agent property values.

Note: Before you set the properties, see *OpenEdge Replication: User Guide* for important background information about OpenEdge Replication and its configuration.

The server properties are divided into the following categories:

- General
- Advanced

To set the OpenEdge Replication Server properties:

1. Click **Resources** in the management console menu. All resources managed by your console appear in the grid frame.
2. Filter or search for the database whose Replication Server properties you want to set.
3. Click the Edit icon. The **Database** details page appears.
4. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
5. In the **OpenEdge Replication Group Links** section, click **Server**. The **Database Replication Configuration** page is displayed.
6. Set the following **General** properties:

   - **Database**—Specifies the source database name.
   - **Defer agent startup (minutes)**—Specifies for how long, in minutes, the server attempts to connect to an agent if the first connection attempt is unsuccessful. The minutes value must be greater than or equal to zero and less than or equal to 10080.
   - **Keep alive (seconds)**—Specifies a timeout period for communications between a server and its agents. If a connection between the server and the agent is not verified before the timeout expires, then failure recovery begins.

     By default, this property is enabled and has a value of 300 seconds. The minimum value is 60 seconds. There is no maximum value.
   - **Server transition timeout (seconds)**—Specifies the number of seconds the target database waits before it performs auto-transition. This property is ignored when transition is set to manual.

     The value is incremented by the sum of the connect-timeout values for all configured agents. The minimum and maximum values are 0 and 86400 seconds, respectively.
   - **Server transition**—Specifies how to transition the target database to a normal database.

     The following occurs if a synchronous agent or an asynchronous critical agent cannot reconnect with the source database within the time specified in the transition-timeout property:

     - **auto**—The agent automatically transitions the target database.

- **manual**—The agent listens for the server until it reconnects with the source database or the DSRUTIL command is executed. The `DSRUTIL db-name -C` transition command transitions the target database.

7. Set the following **Advanced** properties:

   - **Other arguments**—Any other arguments for which there is no defined property available.
   - **Schema lock action**—Specifies the action an agent takes if an exclusive schema lock is not granted. Possible actions are:

     - **wait**—The agent waits until the exclusive schema lock is granted. The server blocks until the exclusive schema lock is granted.
     - **force**—The agent makes up to 5 attempts to acquire the exclusive schema lock. If the fifth attempt fails, then the agent disconnects all users from the target and makes another attempt. If the last attempt fails, then the server and all agents terminate. When schema update activity completes, the server and target can be restarted.

   - **Increment polling delay (milliseconds)**—Specifies the value, in milliseconds, to increase the polling delay.

     By default, the polling delay increment is 5 ms. Set milliseconds to a value between 1 and 1000, inclusive. The value must be less than **Maximum polling delay**.

   - **Polling delay minimum (milliseconds)**—Specifies the minimum value, in milliseconds, for a polling delay. By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

     The value must be greater than or equal to one and less than or equal to 10.

   - **Polling delay maximum (milliseconds)**—Specifies the maximum value, in milliseconds, for a polling delay. By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

     The value must be greater than 500 and less than 1000.

   - **Server port minimum**—The minimum port available.
   - **Server port maximum** The maximum port available.
   - **Agent shutdown action**—Specifies the action an agent takes during a shutdown when the Replication server ends. Possible actions are:

     - **recovery**—The agent remains active but in a standby state waiting for the replication server to reconnect.
     - **normal**—The agent terminates. The target database continues to run.

8. Click **Save**.

Set OpenEdge Replication Agent properties

You set OpenEdge Replication Agent properties to define the configuration for the local agent running for the target database.

Note: For an introduction to OpenEdge Replication and its configuration, see *OpenEdge Replication: User Guide*.

To set the OpenEdge Replication Agent properties:

1. Click **Resources** in the management console menu. All resources managed by your console appear in the grid frame.
2. Filter or search for the database whose Replication Agent properties you want to set.
3. Click the Edit icon. The **Database** details page appears.
4. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
5. In the **OpenEdge Replication Group Links** section, click **Agent**. The **Database Replication Configuration** page opens.
6. Set the following properties:

   - **Agent name**—Specifies the OpenEdge Replication agent name. By convention, the agent name should match the name specified in the source database properties file.

     ALL is not allowed as an agent name. Each agent must have a unique name.

   - **Database**—Specifies the source database name.
   - **Connection Timeout (seconds)**—Specifies how many seconds the OpenEdge Replication agent waits for a connection from the OpenEdge Replication server before the Replication agent shuts itself down.

     The seconds value must be greater than or equal to 120, and less than or equal to 86,400.

     Using this property means you do not have to do a forced shutdown on your target database. If the OpenEdge Replication agent does not receive a connection attempt from the OpenEdge Replication server before the number of seconds specified have elapsed, then the OpenEdge Replication agent terminates and allows a limited number of system-level target database connections.

   - **Polling delay minimum (milliseconds)**—Specifies the minimum value, in milliseconds, for a polling delay.

     By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

     The milliseconds value must be greater than or equal to one, and less than or equal to ten.

   - **Polling delay maximum (milliseconds)**—Specifies the maximum value, in milliseconds, for a polling delay.

     By default, the polling delay starts at 5 milliseconds and automatically increases during periods of inactivity to a maximum of 500 milliseconds.

     The milliseconds value must be greater than or equal to 500 and less than or equal to 1000.

   - **Agent minimum port**—Specifies the minimum TCP port number. The agent selects a port in a range between the values specified by the minimum port and the maximum port.

     The minimum port number value must be greater than 1024 and less than the value of the maximum port.

   - **Agent maximum port**—Specifies the maximum TCP port number for the agent.

     The maximum port number value must be greater than the minimum port + 1, and less than the maximum allowable port number on the system.

     For UNIX the maximum port number is 65534. For Windows the maximum port number is 32765.

   - **Other arguments**—Any other arguments for which there is no defined property available.

7. Click **Save**.

Set OpenEdge Replication Transition properties

You set the Transition properties to specify the transition process values, which are used when the target database becomes a normal database after a failure.

Note: Before you set the properties, see *OpenEdge Replication: User Guide* for important background information about OpenEdge Replication and its configuration.

The transition properties are divided into the following categories:

- General
- Start Up
- After Image
- Back Up

To set the OpenEdge Replication Transition properties:

1. Click **Resources** in the management console menu. All resources managed by your console appear in the grid frame.
2. Filter or search for the database whose Replication Transition properties you want to set.
3. Click the Edit icon. The **Database** details page appears.
4. In the **Command and control** section of the page, click **Configuration**. The **Database Configuration** page appears.
5. In the **OpenEdge Replication Group Links** section, click **Transition**. The **Database Replication Configuration** page opens.
6. Set the following **General** properties:
   - **Database role**:
     - **reverse**—The role of the database is reversed: A source database becomes a target database, and a target database becomes a source database.
     - **normal**—The role of the database becomes that of a normal database. The database is no longer enabled for replication after the transition is performed. This is the default value.
   - **Responsibility**:
     - **primary**—This is the primary database.
     - **secondary**—This is the secondary database.
   - **Transition to agents**—Transitions to the first Replication agent in the list when a failure occurs. Transitions to the second Replication agent in the list if the first agent is not available.

     You can specify any configured Replication agent name as a valid value for this property. If you are specifying more than one value, then separate the names by a comma. For example, `agent1,agent2`.
   - **Transition trigger enabled**—Select to view the status (enabled or disabled) of the Transition trigger.
   - **Replication set**—Specifies whether source and target replicas transition together. By default, the source and targets do not transition together. Select to ensure that source and targets transition together.
7. Set the following **Start Up** properties:

- **Restart after transition**—The database can be automatically restarted after transition is performed. When you select this option, the `*-startup-arguments` properties must be supplied, or the database startup will fail:
  - If the database role is normal, then you must specify the `normal-startup-arguments`.
  - If the database role is reversed, then you must specify the `source-startup-arguments` and the `target-startup-arguments`.
- **Source startup arguments**—If the database is transitioned to a source database, then these arguments are used when the database is started. The arguments are appended to the PROSERVE command, which is used to start the database.

  In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example, `-pf` *`db-name`*`.source.pf`.

  Because the database is a source, you must also specify the `-DBService replserv` argument as an indication to the broker to start the Replication server.
- **Target startup arguments**—If the database is transitioned to a target database, then these arguments are used when the database is started. The arguments are appended to the PROSERVE command.

  In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example, `-pf db-name.target.pf`.

  Because the database is a target, you must also specify the `-DBService replagent -S` *`port-number`* `or` *`service-name`* arguments, as an indication to the broker to start the Replication agent and to listen on the TCP/IP port specified with `-S port-number` or `service-name`.
- **Normal startup arguments**—If the database is transitioned to a normal database, then these arguments are used when the database is started. The arguments are appended to the PROSERVE command.

  In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example, `-pf` *`db-name`*`.normal.pf`.
- **Start secondary broker**—The second broker is automatically started. Valid values for the property are `0` and `1`. When the property is set to `1`, the Secondary startup arguments properties must be supplied, or the secondary broker startup will fail:
  - If the database role is normal, then you must specify the **Normal secondary broker startup arguments**.
  - If the database role is reversed, then you must specify the **Source secondary broker startup arguments** and the **Target secondary broker startup arguments**.
- **Source secondary broker startup arguments**—If the database is transitioned to a source database, then these arguments are used when secondary broker is started. The arguments are appended to the PROSERVE command, which is used to start the secondary broker.

  In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example: `-pf sourcesecbrkr.pf`
- **Target secondary source broker startup arguments**—If the database is transitioned to a target database, then these arguments are used when secondary broker is started. The arguments are appended to the PROSERVE command, which is used to start the secondary broker.

  In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example: `-pf targetsecbrkr.pf`
- **Normal secondary broker startup arguments**—If the database is transitioned to a normal database, then these arguments are used when secondary broker is started. The arguments are appended to the PROSERVE command.

In most cases, the only argument specified here should be `-pf` followed by a parameter file name. For example: `-pf normalsecbrkr.pf`

8. Set the following **After image** properties:

   - **Begin after imaging after transition**—Select the option to automatically begin after-imaging (AI) after a target-to-source database transition. AI can be started for a database that has AI areas.
   - **Automatically add after imaging extents**—This property instructs the transition process to automatically add AI areas to the database if:

     - It is transitioned to a source database.
     - There are currently no AI areas for the database.

     AI areas are not added to the database if you do not select the option. If you do select the option, then AI areas are automatically added to the database by using the structure file specified in the `ai-structure-file` property.
   - **Structure file containing after image area definitions**—The name of the structure file that contains the list of AI areas to add.

9. Set the following **Back Up** properties:

   - **Backup method**—Specifies the backup method performed before AI is enabled. The following values are valid:

     - **mark**—Marks the database backed up by using the `rfutil db-name -C mark backedup` command. Marking the database as backed up does not allow future AI extents to be used when recovering from a disaster
     - **full off-line**—Backs up the database offline by using the OpenEdge PROBKUP utility. The backup is performed in two steps. The first backup is a full backup, which is performed before AI is enabled for the database. The second backup is an incremental backup, which is performed after AI is enabled, and after the role of the database is changed.
     - **full on-line**—This backup is performed after the database is restarted, which occurs after the database has been transitioned.

   - **Full on-line and off-line backup arguments**—The arguments required for the full online and offline backups that are performed for the database. At a minimum, the target file or device must be specified in these arguments for both online and offline backups.

     To avoid overwriting a backup, do not use the same target file or device for both the backup and the incremental backup.

     Do not use backup validation parameters, such as `-vp` and `-vf`.

     The arguments should begin with `device-name`.
   - **Incremental backup arguments**—The arguments required for the offline incremental backup performed after AI is enabled and the database's role is reversed.

     The arguments should begin with `device-name`.

Configure parameters and limits for Progress Application Server for OpenEdge

You can configure tuning parameters and process limits of your OpenEdge ABL application using the **Application configuration for <ABL application name>** page.

To set tuning parameters and process limits:

1. Click **Resources** > **Go to Resources** in the OpenEdge Management console menu.

   All resources managed by your console appear in the grid frame.

2. Filter or search for, and then select, the required PAS for OpenEdge instance.

   For example, enter the default PAS for OpenEdge instance name, `oepas1`, in the **Filter** field, or select **Progress Application Server** as the selection in the **Type** drop-down menu and select the PAS instance.

   The **Progress Application Server: <instance name>** page appears.

3. Click an ABL application in **ABL Applications** section.

   The **ABL Application: <ABL application name>** page appears.

   Note: The OpenEdge manager web application, `oemanager`, must be installed on PAS for OpenEdge for you to manage OpenEdge ABL applications and transport services while the PAS for OpenEdge instance is running.

4. Click **Configuration**.

   The **Application configuration for <ABL application name>** page appears.

5. Select the **Performance Tuning and Limits** tab to set the following performance tuning and process limits:

   Table 1: Tuning and process limits parameters

Parameter	Default value	Description
Minimum number of agents	0	Specifies the minimum number of multi-session agents expected at any time. If the number of agents drops below this number, a client request will cause the session manager to start enough agents to equal this number.
Maximum number of agents	10	Specifies the maximum number of concurrent multi-session agents that can run on the AppServer.
Initial number of agents	1	Specifies the number of multi-session agents to start when AppServer is started.
Maximum connections per agent	16	Specifies the maximum number of TCP connections between session manager and multi-session agent.
Maximum ABL sessions per agent	200	Specifies the maximum number of concurrent sessions a multi-session agent can run concurrently.

Parameter	Default value	Description
Maximum number of agents to start	0	Determines the maximum number of agents that may be started at the same time. No limit is enforced if this value is zero or not defined.
Request wait timeout	15000	Specifies the maximum amount of time, in milliseconds, that a request waits in the session manager for a connection to be available before rejecting the request.
Connection wait timeout	3000	Specifies the maximum amount of time, in milliseconds, that the session manager waits for a connection to be freed up before creating a new connection.
Idle agent timeout	300000	Timeout value in milliseconds for an idle agent. If an agent is idle for more than the specified timeout value, then the session is deleted when idleResource cleanup is done.
Idle connection timeout	300000	Timeout value in milliseconds between an AppServer client and the Session Manager. The SessionManager terminates the connection by automatically disconnecting the connection from the AppServer if a connection is idle for more than the specified timeout value.
Idle resource timeout	0	Timeout value in milliseconds to determine the frequency with which the PAS for OpenEdge instance checks for idle resources. Any resource (for example, connection, agent, or client session) that has not been accessed more recently than the specified timeout for that property is terminated. Idle resource checking is disabled if this property is set to zero.
Idle session timeout	300000	Specifies the timeout value in milliseconds for an idle SessionManager session. If a SessionManager session is idle for more than the specified timeout value, then the session is deleted when idleResource cleanup is done.
Agent listener timeout	10000	Specifies the maximum amount of time, in milliseconds, that the session manager waits for a multi-session agent to start on connect.

6. Click:
   - **Save** to save the changes.
   - **Reset** to apply the last-saved settings.

- **Cancel** to reject the recent changes.

Install other web applications shipped with Progress Application Server for OpenEdge

You can install the web applications that are shipped with PAS for OpenEdge from the **Progress Application Server: <instance name>** page.

To install other web applications shipped with PAS for OpenEdge:

1. Click **Resources** > **Go to Resources** in the OpenEdge Management console menu.

   All resources managed by your console appear in the grid frame.

2. Filter or search for, and then select, the required PAS for OpenEdge instance.

   For example, enter the default PAS for OpenEdge instance name, `oepas1`, in the **Filter** field or select **Progress Application Server** as the selection in the **Type** drop-down menu and select the instance.

   The **Progress Application Server: <instance name>** page appears.

3. To deploy a web application. Click **Deploy** on the Web Application tab. The **Deploy web application** dialog appears. In the **Select application .war** file field, perform one of the following actions to upload a web application.

   a) Enter a file name to select the web application `.war` file from the `extras` folder. The auto suggest list appears as you type the name. For example:

      - `oemanager`
      - `oeabl`
      - `manager`
      - `host-manager`

   b) Use the **Browse** option to locate and select an `application.war` file. Type a name for the selected application in the **Web application name** field. The name must contain only alphanumeric characters. It must not contain blank spaces or special characters, except for the underscore (_) and hyphen (-).

   Note: If your security model is set to **Developer**, then `oemanager` and `manager` are deployed by default while creating the PAS for OpenEdge instance.

4. Click **Deploy** to deploy the application on PAS for OpenEdge. The OpenEdge web application is deployed. For web applications that are already installed on the PAS for OpenEdge instance, a **Deployment failed** notification is displayed.

Configure database environment variables

In OpenEdge 11.7.5, you can use OpenEdge Management to configure environment variables for your managed databases. You can also modify the environment variables later, in which case new values override previous values.

To configure a database environment variable in OpenEdge Management:

1. Navigate to the default database configuration page of your database.
2. Select **Environment variables** on the left panel.
3. Click **Edit**.
4. Enter the variable as a name-value pair in the Environment field.
5. Click **Save**.

PAS for OpenEdge

OpenEdge Release 11.7.5 Service Pack adds support for the PAS for OpenEdge features described in the following sections:

- **Manage security properties using SECPROP**—A new Proenv utility called SECPROP enables you to manage security properties in the `oeablsecurity.properties` file.
- **Change to metrics returned by `getMetrics`**—New session manager queue metrics have been introduced that enable you to measure wait times for client requests.

Manage security properties using SECPROP

SECPROP is a command-line utility for managing properties in the `oeablSecurity.properties` file.

The oeablSecurity.properties file

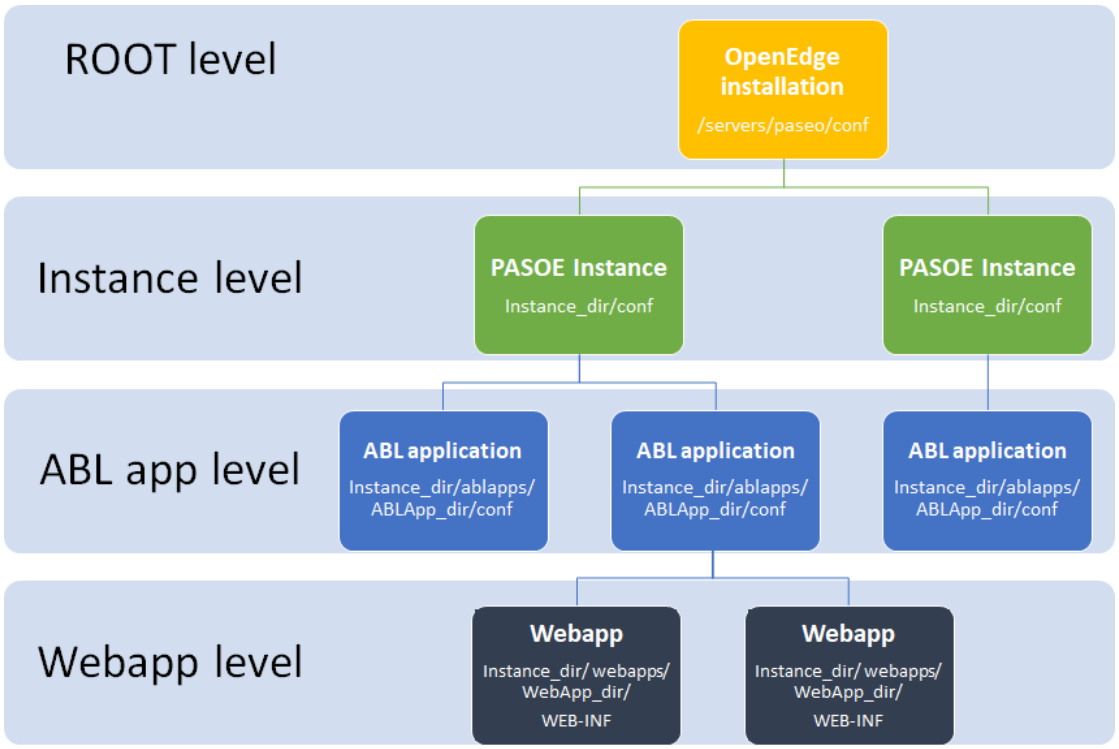
The `oeablSecurity.properties` file provides an easy way for you to configure security properties for a PAS for OpenEdge web application. It acts as an interface to the Spring Security framework that guards the web application. Instead of changing Spring bean constructor and property settings in multiple XML files, you can manage all security properties for your web application from one location in the `oeablSecurity.properties` file.

Security property levels

Even though the purpose of the `oeablSecurity.properties` file is to secure a PAS for OpenEdge web application, the security properties are distributed and replicated across four hierarchical levels:

- In *`OpenEdge_Install_Directory`*`/servers/pasoe/conf`: When you install OpenEdge, a copy of the `oeablSecurity.properties` file is created in this directory. This is a master file that contains all security properties that can be set for a web application.
- In *`PASOE_instance_directory`*`/conf`: When you create a PAS for OpenEdge instance, a copy of the `oeablSecurity.properties` file is created in this directory from the master file. This file inherits a subset of the master file's properties.
- In *`PASOE_instance_directory`*`/ablapps/ABLApp_directory/conf`: When you deploy an ABL application to a PAS for OpenEdge instance, a copy of the `oeablSecurity.properties` file is created in this directory. It inherits properties from the instance-level, security properties file.
- In *`PASOE_instance_directory`*`/webapps/WebApp_directory/WEB-INF`: When you deploy a web application for an ABL application to a PAS for OpenEdge instance, a copy of the `oeablSecurity.properties` file is created in this directory. It inherits properties from the ABL application-level, security properties file.

The inheritance architecture is represented in the following diagram:



A parent-child relationship exists between each level in this architecture. If you modify security properties at the parent level, then the changes are propagated to any new children.

For example, if you modify the `oeablSecurity` file for a PAS for OpenEdge instance, then any new ABL applications that are deployed to the instance get a copy of the updated file. Similarly, if you modify security properties at the ABL application level, then any new web applications that are created for that ABL application get an updated copy of the ABL application's `oeablSecurity` file.

Note: You cannot use the SECPROP utility to modify the `oeablSecurity.properties` file at the top-most level; that is, in the *`OpenEdge_installation`*`/servers/pasoe/conf` directory. Furthermore, if you modify properties at any of the child levels, then any changes to the parent-level properties apply only to new children of the same parent.

Syntax

```
secprop log_level(Optional) operation_type_parameter filter operation
```

Log level

Use the following parameters listed to optionally set the logging level. By default, log messages are written to the command-line console. However, you can modify this behavior to write the log output to a file by configuring the `secproutil-logging.xml` file that you can find in the `conf` folder in a PAS for OpenEdge instance's directory, and also in the `/servers/pasoe/conf` directory in the OpenEdge root installation folder.

`-v`

Sets the logging level to INFO. This is the default level. Typically the -v parameter returns a single line log message for each operation performed or error thrown.

`-g`

Sets the logging level to DEBUG. The -g parameter returns debug messages about the operation that was performed or attempted and the files that were accessed. It also includes INFO-level messages.

`-t`

Sets the logging level to TRACE. The -t parameter detailed messages about each method that was called internally while attempting to perform an operation. It also includes DEBUG-level and INFO-level messages.

Operation type parameters

`-h/-help`

Displays the syntax and parameter descriptions for the SECPROP command.

`-m/-manage`

Performs list, update, add, or delete operations.

`-f`

This parameter allows you to merge an existing `oeablSecurity.properties` property file with any level of an instance, such as instance level, ABL application level, and web application level. You must specify `-f` option to use the merge functionality. This functionality simplifies the procedure to add or update properties in the `oeablSecurity.properties` file.

For example, to merge the `myoeablSecurity.properties` file with the `client.login.model=saml` property, perform the following:

Note: The contents of `myoeablSecurity.properties` file should be `client.login.model=saml`.

To merge this file with an instance, type:

```
secprop -f -i oepas1.oepas1.ROOT myoeablSecurity.properties
```

To view the update for the instance, ABL application, and web application, type:

```
secprop -m -iaw oepas1.oepas1.ROOT client.login.model
```

You can view the output as:

```
i: client.login.model=saml
a: client.login.model=anonymous
w: client.login.model=anonymous
```

`-r`

Resolves a property value across hierarchical levels. The SECPROP utility attempts to find the property in the `oeablSecurity.properties` file for a specified web application, ABL application, or PAS for OpenEdge instance. If the utility does not find a properties file at the specified level, it searches in the parent-level security properties file, and then at the grandparent-level, and so on, until it reaches the root installation level. If the property is not found at any of the levels, then an error is displayed.

Filters

`-c`

List the properties in the master `oeablSecurity.properties` file.

`-i` *`PASOE_instance_name`*

The name of a PAS for OpenEdge instance, for example, `-i oepas1`. The operation that you perform (listing, updating, adding, or deleting security properties) applies only to this PAS for OpenEdge instance.

`-a` *`PASOE_instance_name.ABLApp_name`*

The name of an ABL application, for example, `-a oepas1.myABLApp`. The operation that you perform (listing, updating, adding, or deleting security properties) applies only to this ABL application.

`-w` *`PASOE_instance_name.ABLApp_name.Webapp_name`*

The name of a web application, for example, `-w oepas1.myABLApp.myWebApp`. The operation that you perform (listing, updating, adding, or deleting security properties) applies only to this web application.

You can also concatenate filter parameters to perform operations at multiple levels simultaneously.

`-ia` *`PASOE_instance_name.ABLApp_name`*

For example, `-ia oepas1.myABLApp`. You can define operations (listing, updating, adding, or deleting security properties) that apply to the PAS for OpenEdge instance and the ABL application.

`-iaw PASOE_instance_name.ABLApp_name.Webapp_name`

For example, `-iaw oepas1.myABLApp.myWebApp`. You can define operations (listing, updating, adding, or deleting security properties) that apply to the PAS for OpenEdge instance, the ABL application, and the web application.

Read operations

To read the value of a property, enter the property name, for example:

```
secprop -m -i oepas1 http.all.realm
```

To list all the properties of an `oeablSecurity.properties` file, use the `all` switch, for example:

```
secprop -m -i oepas1 -all
```

To read the value of a property at multiple hierarchical levels, combine `i`, `a`, and `w` filter parameters, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp http.all.realm
```

You can also read the values of different properties at different levels, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp i:http.all.realm a:OEClientPrincipalFilterKey
 w:OECORSFilter.allowAll
```

To find the value of a property or to check whether the property exists in an `oeablSecurity.properties` at any of the parent levels of the specified web application, ABL application, or PAS for OpenEdge instance, use the `-r` operational type parameter, for example:

```
secprop -r oepas1.myABLApp.myWebApp w:OECORSFilter.allowAll
```

To find the value of different properties, use the `-r` operational type parameter, for example:

```
secprop -r oepas1.myABLApp.myWebApp i:http.all.realm a:OEClientPrincipalFilterKey
w:OECORSFilter.allowAll
```

Update operations

To update the value of a property, enter the property name and value pair, for example:

```
secprop -m -i oepas1 http.all.realm=OpenEdge
```

To update the value of a property at multiple levels, combine `i`, `a`, and `w` filter parameters, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp http.all.realm=OpenEdge
```

You can also update the values of different properties at different levels, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp i:http.all.realm=OpenEdge
a:OEClientPrincipalFilterKey=myKey w:OECORSFilter.allowAll=false
```

Note: You can update a property only if the property already exists in the `oeablSecurity.properties` file that you are trying to modify. If the property does not exist, then the update operation fails for that file and the SECPROP utility generates an error.

Add operations

To add a property, use the + switch in conjunction with a filter parameter, for example:

```
secprop -m -i oepas1 +i:http.all.realm=OpenEdge
```

To add a property at multiple levels, combine the `i`, `a`, and `w` filter parameters, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp http.all.realm=OpenEdge
```

You can also add different properties at different levels by using filter parameters:

```
secprop -m -iaw oepas1.myABLApp.myWebApp +i:http.all.realm=OpenEdge
+a:OEClientPrincipalFilter.domain=progress +w:OECORSFilter.allowAll=true
```

Note: You can add a property only if the property does not already exist. If the property exists in an `oeablSecurity.properties` file, then the add operation fails for that file, and the SECPROP utility generates an error.

Delete operations

To delete a property, use the ~ switch in conjunction with a filter parameter, for example:

```
secprop -m -i oepas1 ~i:http.all.realm
```

To delete a property at multiple levels, combine the `i`, `a`, and `w` parameters, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp http.all.realm
```

You can also delete different properties at different levels, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp ~i:http.all.realm
~a:OEClientPrincipalFilter.domain ~w:OECORSFilter.allowAll
```

Performing multiple operations

You can perform multiple operations in the same SECPROP command, using the relevant parameters and switches, for example:

```
secprop -m -iaw oepas1.myABLApp.myWebApp ~i:http.all.realm
+a:OEClientPrincipalFilter.domain=progress w:OECORSFilter.allowAll
```

Change to metrics returned by getMetrics

PAS for OpenEdge is designed to support many clients with a comparatively small quantity of server resources. It does this by directing incoming client requests to run on various shared ABL sessions, which run on one or more multi-session agents. Additional sessions are created if the volume of incoming requests exceeds the capacity of existing ABL sessions, up to a limit that may be configured. If the configured maximum number of sessions is reached, then the session manager queues additional requests and allocates them to ABL sessions as they become available. This allows the server to handle more client requests, with fewer ABL sessions, but at the possible expense of speed. Consequently, to determine how well the configured system handles client workload, it is useful to measure the number of requests that must be queued in the session manager.

In OpenEdge releases prior to 11.7.4, you could use the `getMetrics` operation in OpenEdge JMX to return a collection of data regarding the size of these request queues for a specific ABL application, as the following metrics:

Table 2: Queue metrics prior to 11.7.4

The following metric . . .	… measured
`timesQueued`	The total number of requests that were queued.
`avgQueueDepth`	The average number of requests that were queued.
`maxQueueDepth`	The maximum number of requests that were queued.

Unfortunately, this approach was not optimal because the truly useful metric is not **how many** requests are queued by the session manager for a specific ABL application, but rather **how long** the requests must wait before being executed.

In OpenEdge 11.7.4, the session manager queue metrics have been replaced by the following, which measure the wait times for requests:

Table 3: Metrics in 11.7.4 and later

The following metric . . .	. . . measures for a specific ABL application
`numReserveABLSessionWaits`	The total number of times that a client request waited for an ABL session to become available.
`avgReserveABLSessionWaitTime`	The average time (in milliseconds) that a client request waited for an ABL session to become available or to be created. This average excludes requests that were immediately allocated to ABL sessions without waiting (that is, “zero wait time”). This field is `totReserveABLSessionWaitTime` divided by `numReserveABLSessionWaits`.
`maxReserveABLSessionWaitTime`	The maximum time (in milliseconds) that a client request waited for an ABL session to become available or to be created.
`totReserveABLSessionWaitTime`	The total amount of time all client requests waited for an ABL session to become available. (This value is not critical to understanding overall performance, but is required to complete the average.)

You can observe this change in the collection of data returned by the `getMetrics` operation and make adjustments to improve the handling of client workload for a specific ABL application.

Progress Developer Studio for OpenEdge

Release 11.7.5 contains Progress Developer Studio for OpenEdge updates to provide the following features:

- **PCT version change**—PCT (Progress Compilation Tools) is a set of ANT libraries, designed to automate work in OpenEdge environment. It helps to automate build generation in continuous integration process for OpenEdge applications. The upgraded version is 211.
- **Telerik UI for WinForms controls upgrade**—The Telerik UI for WinForms controls that installs by default is upgraded to R1 2019 (2019.1.115.40). If you need a different version, then you can download it from the Telerik web site.

Security updates

OpenEdge continues to update its OpenSSL library to pass the latest common vulnerabilities and exposures (CVE). In OpenEdge 11.7.5, OpenSSL has been upgraded to 1.1.1. As with previous OpenSSL versions, the TLSv1.2 protocol continues to be the default if a specific version is not specified. As part of the move to OpenSSL version 1.1.1, the newer, more secure TLSv1.3 is now available as an option that you may manually activate.

If you want to enable support for `TLSv1.3` or any other protocol supported by OpenEdge, then set the value of the following environment variables to `TLSv1.3` for the corresponding supported protocol:

- `PSC_SSLCLIENT_PROTOCOLS`
- `PSC_SSLSERVER_PROTOCOLS`

For information about how to set the value of the environment variables for OpenEdge clients and servers, see Change default protocols and ciphers for Progress OpenEdge clients and Change default protocols and ciphers for Progress OpenEdge servers.