Quick Start Guide: Moving your deployed applications to the Progress® Application Server for OpenEdge®

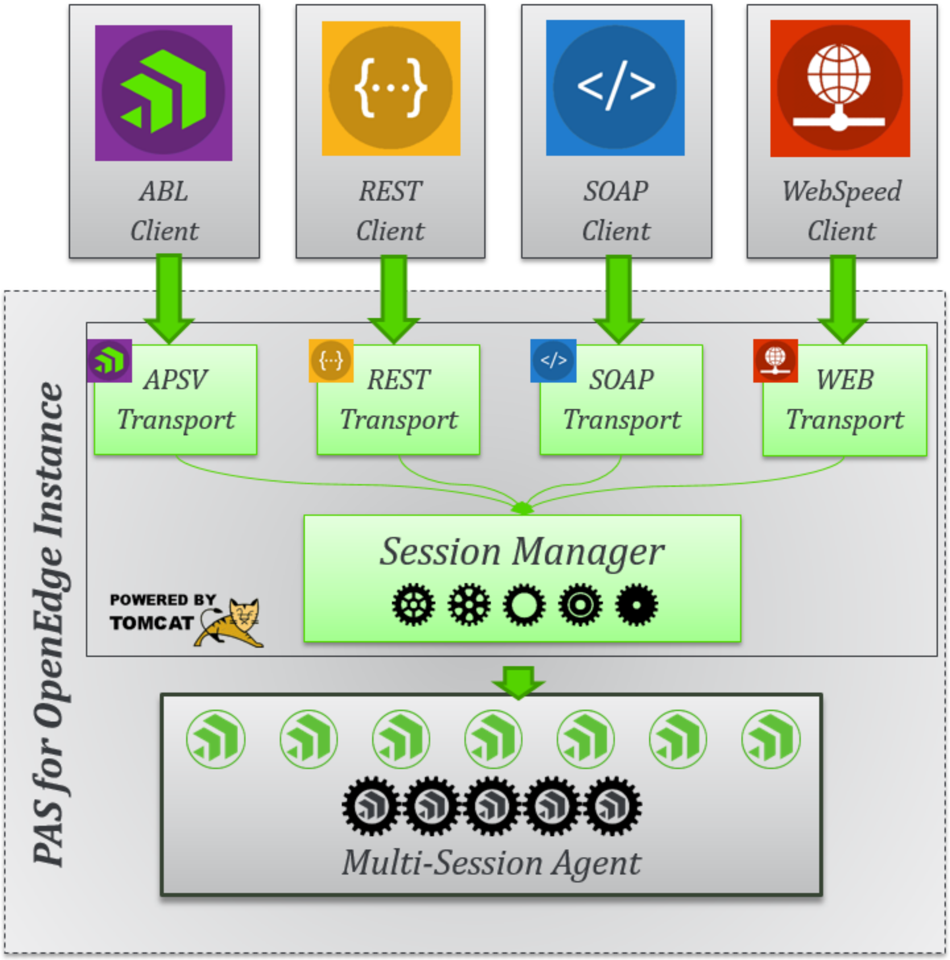
Product Version: OpenEdge 11.7.3 and higher

Table of Contents

Welcome

It's moving day and we are here to help. The *Progress Application Server for OpenEdge (PAS for OpenEdge)*, is a *unified web server.* With PAS for OpenEdge, all client types communicate with your ABL code using one of four transports shown below. The Session Manager then distributes those requests to the underlying Multi-Session Agents to ensure efficient use of your server resources. Built on Apache Tomcat web server technology and enhanced by OpenEdge development to manage the specific needs of your OpenEdge clients, you get the best of both worlds. Apache updates ensure compliance with industry standards like Spring security while OpenEdge development ensures we can handle all client types in a single server with no additional products.

Although many of the configuration parameters appear to be similar, configuring a successful deployment in the new multi-session environment is different. The guidelines presented here are designed to help you with your move.



Moving your deployed applications to the Progress® Application Server for OpenEdge®

To move your applications, you will need to complete the following steps:

1. Install PAS for OpenEdge (Production license)
2. Create a new instance
3. Convert and customize properties files
4. Move your application files
5. Update your client connection parameters
6. Start your PAS for OpenEdge instance

Step 1: Installing PAS for OpenEdge

We offer *development* and *production* licenses for PAS for OpenEdge. You will want to install *both* types since you will need development features to convert files for use on the production instance. The production license implements stronger security and limits external management of your PAS for OpenEdge instances so it is the preferred license for exposing your application to the world.

1. Locate or download *version 11.7.3 or higher software needed for your configuration. Typically, Development and Production licenses for PAS for OpenEdge as well as necessary development and database tools.*
2. If you are less familiar with the installation process, see Unix or Windows installation steps.

Step 2: Creating a new instance

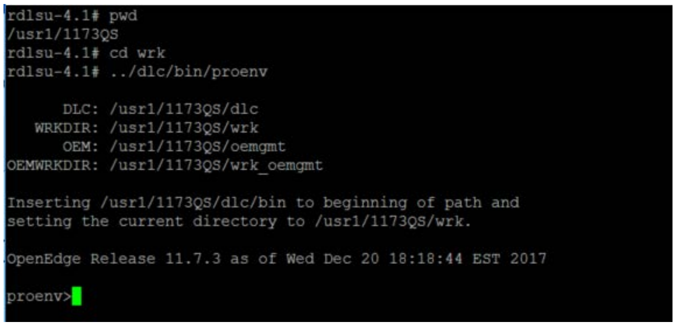
After installing, use *pasman create* to create an instance. This instance will act as the staging area for deploying your existing application to a *production license PAS for OpenEdge* instance before you move to a live environment. Unix commands are provided, using a Windows command prompt will provided similar results.

1. Change to your work directory.

   ```
   >cd wrk
   ```

2. From your working directory, use *proenv* to set the environment variables and paths.

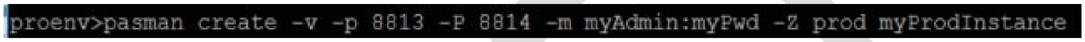
   ```
   >../dlc/bin/proenv
   ```



```
rdlsu-4.1# pwd
/usr1/1173QS
rdlsu-4.1# cd wrk
rdlsu-4.1# ../dlc/bin/proenv

      DLC: /usr1/1173QS/dlc
   WRKDIR: /usr1/1173QS/wrk
      OEM: /usr1/1173QS/oemgmt
OEMWRKDIR: /usr1/1173QS/wrk_oemgmt

Inserting /usr1/1173QS/dlc/bin to beginning of path and
setting the current directory to /usr1/1173QS/wrk.

OpenEdge Release 11.7.3 as of Wed Dec 20 18:18:44 EST 2017

proenv>
```

3. Run *pasman create* to create a named instance.

```
pasman create -v -p 8813 -P 8814 -m myAdmin:myPwd -Z prod
myProdInstance
```



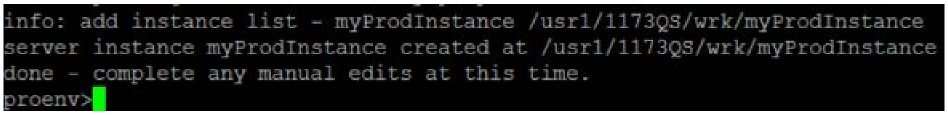
```
proenv>pasman create -v -p 8813 -P 8814 -m myAdmin:myPwd -Z prod myProdInstance
```

The *pasman* command manages and configures PAS instances using a variety of *actions*. The *create* action creates a new instance from a template. It takes several mandatory and optional *options.* On Unix, the command script is *pasman*. On Windows, the command script is *pasman.bat*. Here are the options used for this create.

Option	Description
-v	Verbose output to console provides feedback during the create process.
-p	TCP port that listens for HTTP message. This example uses *8813*.
-P	TCP port that listens for HTTPS message. This example uses *8814*.
-m	uid:pwd is a username and password combination to replace the default tomcat:tomcat found in Apache Tomcat. The example uses *myAdmin:myPwd* for demonstration purposes only.
-Z	Identifies type of instance. The example uses *prod* to lock down server with production server security settings. You may notice that development accounts and transports are disabled by default.
Instance pathname	Absolute path to the directory where instance gets created. The example uses *myProdInstance*

For more information, see the *Progress® Application Server for OpenEdge®: Administration Guide*.

a. Success looks like this.



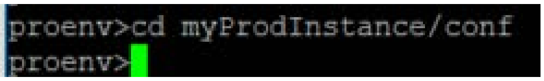
```
info: add instance list - myProdInstance /usr1/1173QS/wrk/myProdInstance
server instance myProdInstance created at /usr1/1173QS/wrk/myProdInstance
done - complete any manual edits at this time.
proenv>
```

Step 3: Converting and customizing properties

With any move, you need to adjust to your new home. In this section, you will use a command line tool to convert existing properties to new properties on your new server.

1. Change to the instance/conf directory.

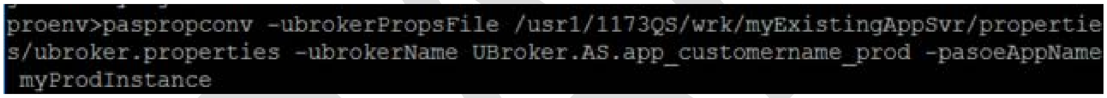
   ```
   proenv>cd myProdInstance/conf
   ```

   ```
   proenv>cd myProdInstance/conf
   proenv>
   ```

2. Run the *paspropconv* conversion tool to take an existing *ubroker.properties* file as input and output a merge file with converted and suggested changes for your new server.

   ```
   proenv> paspropconv -ubrokerPropsFile
   [fullpath]/ubroker.properties
   -ubrokerName UBroker.AS.app_customername_prod -pasoeAppName
   myProdInstance
   ```

   ```
   proenv>paspropconv -ubrokerPropsFile /usr1/1173QS/wrk/myExistingAppSvr/propertie
   s/ubroker.properties -ubrokerName UBroker.AS.app_customername_prod -pasoeAppName
    myProdInstance
   ```

Option	Description
-ubrokerPropsFile	[Required] Path to existing Classic AppServer `upbroker.properties` file. In this example, `/usr1/1173QS/wrk/myExistingAppSrv/properties/ubroker.properties`.
-ubrokerName	[Required] Fully qualified name of the broker you are migrating. In this example, `UBroker.AS.app_customername_prod`.
-pasoeAppName	[Required] Name of the new instance. In this example, `myProdInstance`.

3. Review the *ubrokerName.merge* file. Your recommendations will be specific to your environment and configurations. Consider each recommendation made by the tool before applying them in the next step.

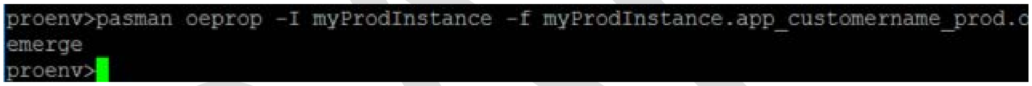
Section	Description
Input arguments	Required and default arguments for `paspropconv`.
Operating modes	Advice on changing existing operating modes to new operating modes.
Startup event procedures	Advice on using new event procedures.
Manual edits	Advice on deployment difference between classic and PAS for OpenEdge systems.

Database connections	Advice on database connections.
Capacity & performance	Advice on properties and their impact on performance.
Redundancy & failover	Advice on `minAgents` and `maxAgent` settings.
Agent port conflicts	Advice on potential port conflicts.
Max # of ABL sessions	Advice on memory management.
Max # concurrent requests	Advice on interactions between Tomcat connector and number of concurrent requests.
SvrStartupParam	Updated property to include default PAS for OpenEdge entries.
PROPATH	Updated to use commas.
Properties to merge	Summary of the properties to merge

4. After you've reviewed and adjusted the merge file to match your system. Run

   pasman oeprop to merge the new properties into the *openedge.properties* file.

```
\\rdlserv\cdimage\OpenEdge\117\nt64\untested\SP_Jan2
myProdInstance.app_customer_prod.oemerge
```



```
proenv>pasman oeprop -I myProdInstance -f myProdInstance.app_customername_prod.c
emerge
proenv>
```

5. Make local copy of the generate **app_customername_prod_setenv[.sh|bat]** in your *instancePath*/**bin** to short your startup scripts.

```
proenv> cp
/usr1/1173QS/wrk/myProdInstance/conf/app_customername_prod_sete
nv.sh /usr1/1173QS/wrk/myProdInstance/bin
```

Important Security Note: The migration tool suggests a PROPATH that allows code to remain in existing directories. However, there are recommended locations in PAS for OpenEdge for application code. These file locations promote security, scaling, extensibility, and deployment.

- For application code, move source code or compiled code to *instancePath*/**webapps/ROOT/WEB-INF/openedge**

- For code common to multiple applications, move source code or compiled code to *instancePath*/**openedge**

Step 4: Moving your code

(NOTE: Please note the following sections are still being revised.)

Since your code will vary please complete the steps appropriate for moving your code, based on the existing code types, for use with new transports.

Moving AppServer code to use APSV transport

If you set up PROPATH in Step 3, then your application code is now available to clients via your PAS for OpenEdge instance.

1. Review the PROPATH entry in *instancePath/conf/ openedge.properties* to make sure that it includes the code for your application.
2. Add any additional entries to PROPATH

If you are migrating a classic state-reset or state-aware application, you will also need to configure special connect and disconnect event procedures. See *Migrating AppServer operating modes* in ***Progress® Application Server for OpenEdge®: Application Migration and Development Guide*** for details.

Moving REST adapter code to use REST transport

Since you are no longer using an adapter, you will need to make the following changes. A classic REST web application .war file cannot be deployed to PAS for OpenEdge. If you have a .war, unzip the archive and deploy the application.paar file or get the file from Progress Developer Studio for OpenEdge by exporting it.

1. Copy the code that supports the REST interface API into the directory instancePath*/webapps/ROOT/WEB-INF/openedge*.
2. Navigate to *instancePath/bin*.
3. Run `deployREST[.sh|.bat] source_descriptor ROOT`

Option	Description
-source_descriptor	[Required] either a PAAR file exported from Progress Developer Studio for OpenEdge or a ZIP file containing Mobile catalog files (or other static files).
-ROOT	ROOT is the target service name

See the OpenEdge web application deployment section of Progress Application Server for OpenEdge®: Administration Guide for more information on deployREST usage.

REST adapter code to REST transport	
Web Service Adapters (WSA)	
WebSpeed®	

Step 5: Updating your client connection parameters

Choose from the steps below to update each type of client application to communicate with your new PAS for OpenEdge instance.

Application changes by client types...	
AIA Adapter	
Change clients of the AppServer Internet Adapter (AIA) to use the APSV transport	From: -URL http://host:port/aia/Aia?AppService=asbroker1 To: -URL http://host:port/apsv
OpenEdge	
Change client connection parameters to use the APSV transport	From: IS THERE AN EXAMPLE TO USE HERE ☺ To: -URL http://host:port/apsv
REST Adapter	
Change client URLs to use the ROOT web application	From: http://host:port/restservice/rest/restresource To: http://host:port/rest/restresource
Web Service Adapters (WSA)	
Change clients of the Web Services Adapter (WSA) to use the SOAP transport	From: -WSDL http://host:port/wsa/wsa1/wsdl?targetURI=urn:CustomerSvc To: -WSDL http://host:port/soap/wsdl?targetURI=urn:CustomerSvc
WebSpeed®	
Change WebSpeed clients to use the WEB transport.	From: http://host:port/cgi/wspd_cgi.sh/. . . To: http://host:port/web/. . . *WSASP, WSISA, NSAPI, and CGIIP messengers are not used with PAS for OpenEdge®.

Learning more about PAS for OpenEdge

PAS for OpenEdge differs fundamentally from the Classic AppServer

Although both PAS for OpenEdge and the Classic AppServer run ABL business applications, the architecture and configuration are fundamentally different. PAS for OpenEdge is a web server that uses special web applications to run ABL code. It has all the behavior and features of a web server. Although many of the configuration parameters appear to be similar, the tools and techniques for a successful deployment in the multi-session environment are different. Refer to the output of the property migration tool for useful information on configuring your instance.

The latest release keeps you up-to-date with security and software updates

In the web environment, tracking the latest security changes is extremely important. The PAS for OpenEdge product is continually being updated with new functionality, bug-fixes, and security patches. Stay current to keep your application secure and stable.

Use property files to change PAS for OpenEdge configuration values

Although PAS for OpenEdge is based on Apache Tomcat®, PAS for OpenEdge has simplified the underlying Tomcat configuration and startup for local and remote administration. Rather than making changes to the conf/server.xml or bin/setenv.[bat|sh] files directly, PAS for OpenEdge uses property files and customer extensible script files. See the Progress® Application Server for OpenEdge®: Configuration Guide.

Check all PAS for OpenEdge log files for errors

A client request traverses multiple PAS for OpenEdge subsystems, any one of which can raise an execution error. The recording of the error will occur in the subsystem specific log file. Always check all log files in the logs directory when investigating execution errors. You will find the log files in the instancePath/logs directory. Refer to the Progress® Application Server for OpenEdge®: Administration Guide.

PAS for OpenEdge may fail to start because of errors in your ABL application

The PAS for OpenEdge multi-session agent log file does distinguish between ERROR, WARNING, or INFO messages. As a result, the PAS for OpenEdge pasoestart command can only report whether the agent OS process has stopped due to startup problems. If pasoestart does not specifically report the startup error, you should manually inspect the agent log file in the instancePath/logs directory to determine what the startup problem was and correct it.

Configure and load test your instance before moving into production

Choosing the right machine image size and optimal PAS for OpenEdge configuration for that machine image plays a key role in moving your ABL application into the production environment. PAS for OpenEdge is optimized for resource consumption and optimal throughput under load. Unlike a classic AppServer or WebSpeed you cannot derive ABL application performance using a single client. Make sure to test your application with the anticipated number of concurrent clients to ensure you do not exceed finite resource limits imposed by the OS, OS processes, networking, and OpenEdge database. See the Progress® Application Server for OpenEdge®: Tuning Guide for details.

Stopping a PAS for OpenEdge instance may take some time

Stopping a PAS for OpenEdge instance may take several minutes in some cases. It may even appear hung. PAS for OpenEdge (Tomcat) has a policy that allows its web applications to finish client requests before stopping. A normal PAS for OpenEdge stop is more of a suggestion rather than a hard stop of all processes. If you want PAS for OpenEdge not to wait for client requests to finish before stopping use the pasman stop command and supply the -F option to force shutdown.

The Apache Tomcat® access log is a great troubleshooting tool

The access log is a great help when troubleshooting client connection problems, maximum response times, client login failures from hackers, and much more. While it does take some small amount of processing time and disk space its information cannot be easily obtained from other sources and can be helpful in monitoring the health of your server and tuning its run-time. See the Apache Tomcat® Configuration Reference.

Apache Tomcat® stuck thread detection is a useful alerting mechanism

The Apache Tomcat® stuck thread detection identifies requests that take a long time to process, which might indicate that something is wrong in the request execution. Stuck thread detection is an alerting mechanism that logs a message for requests that take longer than a configurable amount of time to complete. See the Apache Tomcat® Configuration Reference.

PAS for OpenEdge installations on UNIX have file permissions that only allow the root use

On UNIX, the PAS for OpenEdge production installation and instance tailoring eliminates all access by any user account. After an OpenEdge installation completes its instances are configured to be started, stopped, configured, and monitored by the root user account (required by the OpenEdge installer). If all your OS administration is performed via the root user account, no action needs to take place. If you follow best practices and never use the OS root user account, then you will need to change file permissions for core and instance executables and scripts. Refer to the Progress® Application Server for OpenEdge®: Administration Guide.

Glossary

	Definition
ABL Application	A business application written in ABL and executed by an AVM (ABL Virtual Machine) running in a PAS for OpenEdge instance. A **PAS for OpenEdge Instance** must contain one or more **ABL Applications**.
ABL Session	A single instance of an ABL application's run-time context and local data. In ABL this is conceptually represented by the SESSION system handle. One **Multi-Session Agent** has one or more **ABL Sessions.**
ABL Session Manager	An internal component of PAS for OpenEdge that manages client requests and routes them to the appropriate **ABL session** for execution in an **AVM**. An **ABL Application** has one, and only one, **ABL Session Manager.** One **ABL Session Manager** coordinates requests for one or more **Multi-Session Agents.**
ABL Virtual Machine (AVM)	A runtime engine that executes ABL in the appropriate **ABL Session**.
Apache Tomcat®	A commercial web server that provides a web application container, Spring security, and other common web application technologies. PAS for OpenEdge uses Tomcat as its server platform.
Multi-Session Agent	An OS process that is started outside of the PAS for OpenEdge server by the ABL Session Manager. This OS process executes the **_mproapsv** binary executable, which hosts pools of **ABL Sessions**, **AVMs**, and shared resources for executing concurrent ABL requests to an ABL application. One or more **Multi-Session Agents** are coordinated by a single **ABL Session Manager**.
OpenEdge Web Application	A Java web application that is preconfigured with security and Java Servlet interfaces. It translates HTTP requests into ABL requests and schedules execution using an **ABL Session Manager**. In PAS for OpenEdge the default ROOT application is a deployment of oeabl.war, the **OpenEdge Web application**. An **ABL Application** contains one or more **Web Applications.**
Progress Application Server (PAS)	A Progress web server platform which includes administration and deployment tools. PAS is preconfigured to operate as a production ready server. PAS is based on Apache Tomcat®.
Progress Application Server for OpenEdge (PAS for OpenEdge)	A PAS server prepackaged with support for business applications written in ABL. PAS for OpenEdge provides HTTP, REST, WEB and SOAP client access to ABL business applications.
Spring Security	A security framework integrated with PAS for OpenEdge and available to each OpenEdge Web Application, which supplies a full range of direct user login, SSO, and URL authorization services.
Transports	A set of protocol handlers in PAS for OpenEdge. In addition, support for HTTP/HTTPS client access is included in the default ABL (root) application. The transports include SOAP, REST, APSV (equivalent to AIA), and WEB (WebSpeed and WebHandler) access. Support for these transports is built-in and does not need to be installed separately.

Web Application	A collection of servlets, html pages, classes, and other resources that canbe bundled and run in a container. PAS, which is built on Apache Tomcat®, is a web application container.

Term	Definition
ABL Application	A business application written in ABL and executed by an AVM (ABL Virtual Machine) running in a PAS for OpenEdge instance. A **PAS for OpenEdge Instance** must contain one or more **ABL Applications**.
ABL Session	A single instance of an ABL application’s run-time context and local data. In ABL this is conceptually represented by the SESSION system handle. One **Multi-Session Agent** has one or more **ABL Sessions.**
ABL Session Manager	An internal component of PAS for OpenEdge that manages client requests and routes them to the appropriate **ABL session** for execution in an **AVM**. An **ABL Application** has one, and only one, **ABL Session Manager.** One **ABL Session Manager** coordinates requests for one or more **Multi-Session Agents.**
ABL Virtual Machine (AVM)	A runtime engine that executes ABL in the appropriate **ABL Session**.
Apache Tomcat®	A commercial web server that provides a web application container, Spring security, and other common web application technologies. PAS for OpenEdge uses Tomcat as its server platform.
Multi-Session Agent	An OS process that is started outside of the PAS for OpenEdge server by the ABL Session Manager. This OS process executes the **_mproapsv** binary executable, which hosts pools of **ABL Sessions**, **AVMs**, and shared resources for executing concurrent ABL requests to an ABL application. One or more **Multi-Session Agents** are coordinated by a single **ABL Session Manager**.
OpenEdge Web Application	A Java web application that is preconfigured with security and Java Servlet interfaces. It translates HTTP requests into ABL requests and schedules execution using an **ABL Session Manager**. In PAS for OpenEdge the default ROOT application is a deployment of oeabl.war, the **OpenEdge Web application**. An **ABL Application** contains one or more **Web Applications.**
Progress Application Server (PAS)	A Progress web server platform which includes administration and deployment tools. PAS is preconfigured to operate as a production ready server. PAS is based on Apache Tomcat®.
Progress Application Server for OpenEdge (PAS for OpenEdge)	A PAS server prepackaged with support for business applications written in ABL. PAS for OpenEdge provides HTTP, REST, WEB and SOAP client access to ABL business applications.
Spring Security	A security framework integrated with PAS for OpenEdge and available to each OpenEdge Web Application, which supplies a full range of direct user login, SSO, and URL authorization services.

Transports	A set of protocol handlers in PAS for OpenEdge. In addition, support for HTTP/HTTPS client access is included in the default ABL (root) application. The transports include SOAP, REST, APSV (equivalent to AIA), and WEB (WebSpeed and WebHandler) access. Support for these transports is built-in and does not need to be installed separately.
Web Application	A collection of servlets, html pages, classes, and other resources that can be bundled and run in a container. PAS, which is built on Apache Tomcat®, is a web application container.