Running PAS for OpenEdge in a Production Environment

All “classic” OpenEdge server products (OpenEdge AppServer, WebSpeed Transaction Server) were configured to run in *development mode* by default. Development mode minimizes or eliminates any accessibility restrictions, so a developer can quickly use the product “out of the box” to develop, test, and debug applications.

The newer Progress Application Server (PAS) for OpenEdge also has a development mode product, Progress Development Application Server for OpenEdge (`Progress Dev AS for OE`). In addition, it is available as a *production mode* product, Progress Production Application Server for OpenEdge (`Progress Prod AS for OE`). A production instance of PAS for OpenEdge is not meant to “just work” out of the box. It enforces the current security best practices followed by many other current web-based products. You must make a number of configuration changes before you can run and deploy your applications to a production instance.

The following topics describe configuration and other issues that you should know in order to run the production version of PAS for OpenEdge:

- Configuration tools
- Checking the security mode of the server
- User and file permissions
- Creating a production instance
- Enabling transports
- Enabling WebSpeed support
- Missing and disabled features
- Enabling authentication
- Other security considerations
- Setting environment variables

However, they do not describe security issues in any depth. For more information on security see the PAS for OpenEdge documentation.

Configuration utilities

Although you can edit PAS for OpenEdge configuration files directly, it is a better practice to use configuration utilities to change configuration files. This is especially critical when working with production instances. Manual edits are error prone which can cause unexpected downtime that is difficult to debug and fix.

You can manage configurations with the OpenEdge Management and OpenEdge Explorer tools. (See *OpenEdge Management: PAS for OpenEdge Configuration* for more information.) In addition, there are a number of command line utilities available in PAS for OpenEdge.

To view or make changes in an instance's `appserver.properties` and `catalina.properties` files use the `config` action of the TCMAN command line utility. Run TCMAN from the `/bin` directory of your instance. For example:

- List all the properties in both files:

```
../bin/tcman[.sh | .bat] config
```

- Show the value for a particular property:

```
../bin/tcman[.sh | .bat] config property_name
```

- Set or change the value of a property:

```
../bin/tcman[.sh | .bat] config property_name = value
```

To view or make changes in an instance's `openedge.properties` file you can use the OEPROP command line utility. Run OEPROP from the `/bin` directory of your instance. For example:

- Show the value for a particular property:

```
../bin/oeprop[.sh | .bat]group_name.property_name
```

- Set or change the value of a property

```
../bin/oeprop[.sh | .bat]group_name.property_name = value
```

Checking the security mode of the server

If you don't know which product security model your instance is running (development or production) you can verify by:

- Running *instance_dir*/bin tcman[.sh | .bat] env. The security model is listed in the second to the last line of the output.
- Running *instance_dir*/bin/tcman[.sh | .bat] config psc.as.security.model.
- Check the setting for the psc.as.security.model property in the instance's ../conf/appserver.properties file.

User and file permissions

The UNIX installation procedure of a production version of PAS for OpenEdge sets user and file permissions to limit who can access, manage, and configure the server.

Immediately after installation, all permissions for the PAS for OpenEdge base configuration (a.k.a. $CATALINA_HOME which is set to $DLC/servers/pasoe) is owned by ROOT and is only accessible and executable by ROOT users and groups. Subsequently, you can set up your own groups with user accounts that have permission to run the CATALINA_HOME files. When you create a new instance from CATALINA_HOME while logged in with one of these new user accounts, you will have access to the files on the new instance.

On Windows, no user or group ACLs are set during installation. You should manually set permissions on the %DLC%\servers\pasoe directories, sub-directories and files to prevent unrestricted access.

Creating a production instance

When you install a development version, a default instance is created in your work directory. However, there is no default instance created when you install a production version of PAS for OpenEdge. A default instance implies that ports, name spaces, and other values would be well-known and would result in a security risk. Therefore, before you can deploy an application to PAS for OpenEdge, you must create an instance, usually by running the TCMAN create action from $CATALINA_HOME/bin.

> **Note:** For security reasons, you should never attempt to run the PAS core server in $CATALINA_HOME as a production server or deploy to it. The core PAS is intended only as a source for instance creation.

Enabling transports

In PAS for OpenEdge, transports support client access to deployed web applications via a specific protocol. Currently, there are four available transports, APSV, REST, SOAP, and WEB.

By default, all transports are disabled in the production version of PAS for OpenEdge. Note that all transports are enabled in the development version. On a production instance, you should enable only those transports you are going to use.

To enable a transport, set the transport's `adapterEnabled` property to `1` by editing the instance's /conf/openedge.properties file.

For example, the following snippet shows the APSV transport enabled for an instance named `oepas1`:

```
[oepas1.ROOT.APSV]
    adapterEnabled=1
```

You can also use the OEPROP command line utility to modify the `openedge.properties` file. For example:

```
oeprop oepas1.ROOT.APSV.adapterEnabled=1
```

> Note: You must restart a PAS for OpenEdge instance in order for changes in its `openedge.properties` file to take effect.

Enabling WebSpeed support

Beginning with OpenEdge release 11.6, PAS for OpenEdge includes a WEB transport to support WebSpeed as well other types of web application. Like the other transports, the WEB transport is disabled by default in the production version of PAS for OpenEdge (*all* transports are enabled in development versions.)

The first step to enable WebSpeed support in PAS for OpenEdge is to enable the WEB transport in the `/conf/openedge.properties` file. You can edit the file directly and set `adapterEnabled=1` for the WEB transport, for example (where `oepas1` is the instance name):

```
[oepas1.ROOT.WEB]
    adapterEnabled=1
```

You can also use the OEPROP command line utility to update the `openedge.properties` file to enable the WEB transport. For example (where `oepas1` is the instance name):

```
oeprop oepas1.ROOT.WEB.adapterEnabled=1
```

The next step is to set the appropriate web handler for WebSpeed. Web handlers are control programs that manage the execution of web objects in PAS for OpenEdge instances. They replace the classic WebSpeed control program, `web-disp.p`, which cannot be migrated to PAS for OpenEdge. You can modify the built-in web handler, or create a new web handler to implement any customizations you may have made to `web-disp.p`. See the *PAS for OpenEdge: Application Migration and Development Guide* for more information.

The default WebHandler for production instances is `OpenEdge.Web.DefaultWebHander`. If you plan on deploying existing WebSpeed applications (i.e. applications developed for the "classic" WebSpeed Transaction Server), use `OpenEdge.Web.CompatibilityHandler` instead of the default. If you do not change it, a 405 error will always be generated in response to a client request.

If you are developing new WebSpeed applications using the newer coding practices tailored for deployment on PAS for OpenEdge, keep the default WebHandler. Any invalid URLs result in a 405 error.

You can edit the `openedge.properties` file to set the web handler. For example (where `oepas1` is the instance name):

```
[oepas1.ROOT.WEB]
                 .
                 .
                 .
     defaultHandler=OpenEdge.Web.CompatibilityHandler
```

You can also use the OEPROP command line utility to update the `openedge.properties` file to the web handler. For example (where `oepas1` is the instance name):

```
oeprop oepas1.ROOT.WEB. defaultHandler=OpenEdge.Web.CompatibilityHandler)
```

> Note: You must restart a PAS for OpenEdge instance in order for changes in its `openedge.properties` file to take effect.

Features disabled in production by default

As a best practice, the production version of PAS for OpenEdge should be as secure as possible after initial installation. To facilitate this, all debugging and management features are disabled by default but can be re-enabled if needed. Some other features are just not included but could be installed or implemented if necessary.

The following is a list of the disabled features in a production version of PAS for OpenEdge:

- **Status home page is not implemented.**

  The development version of PAS for OpenEdge includes a default home page that displays status information. This is provided for debugging purposes so the developer has quick access to server data. In order to prevent internal data from being viewed externally, there is no default status page in the production version.

- **Status responses are off by default**.
  Status response is controlled by the `statusEnabled` property in `openedge.properties`. If enabled, status is returned in a JSON page when you query the transport with the following syntax:

  ```
  http://hostname:portnumber/apsv|rest|soap|web
  ```

- **The Tomcat manager application is not installed.**

  If you wish to have the Tomcat manager running, you need to deploy the `$DLC/servers/pasoe/extras/manager.war` file to your instance.

  > **NOTE:** If you do enable the manager, change the users and passwords from the defaults in `~/conf/tomcat-users.xml`

- **The OpenEdge REST API manager is not installed.**

  If you wish to use the REST API's or OpenEdge Management you need to deploy the `$DLC/servers/pasoe/extras/oemanager.war` file to your instance.

  NOTE: If you do enable the manager change the users and passwords from the defaults in `~/conf/tomcat-users.xml`

Enabling authentication

Before actually running and deploying to a production instance, you should increase the security level by enabling authentication. Out of the box, the authentication mode of the production server is set to anonymous, which is the lowest security level. This is done because it is impossible to foresee which authentication implementation you might choose among the countless processes and products.

The Spring Security framework, which is an integral part of PAS for OpenEdge, is adaptable to a wide variety of authentication models and architecture.

You select which authentication mode to enable in the `instance_name/webapps/ROOT/WEB-INF/web.xml` file. Find the `BEGIN:Spring security.definition` section (near the top of the file). Notice that there is a list of Spring Security XML configuration files. By default, the one that is uncommented is `/WEB-INF/oeablSecurity-anonymous.xml` which implements the anonymous model of authentication which is the lowest level of security. You must comment out the anonymous configuration file and uncomment a file that matches your authentication mode.

The following shows the list of available Spring Security files as they are listed in the web.xml file with the anonymous configuration still selected:

```
<context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>
<!-- USER EDIT: Select which application security model to employ
            /WEB-INF/oeablSecurity-basic-local.xml
            /WEB-INF/oeablSecurity-anonymous.xml
            /WEB-INF/oeablSecurity-form-local.xml
            /WEB-INF/oeablSecurity-container.xml
            /WEB-INF/oeablSecurity-basic-ldap.xml
            /WEB-INF/oeablSecurity-form-ldap.xml
            /WEB-INF/oeablSecurity-basic-oerealm.xml
            /WEB-INF/oeablSecurity-form-oerealm.xml
            /WEB-INF/oeablSecurity-form-saml.xml
            /WEB-INF/oeablSecurity-basic-saml.xml
-->
            /WEB-INF/oeablSecurity-anonymous.xml
        </param-value>
    </context-param>
```

NOTE: when you change the `oeablSecurity-*.xml` file from `local` to another type, adding and changing the users and passwords is not handled by the "local" `~/conf/tomcat-users.xml` file. Also you will need to modify the oeableSecurity*.xml file to access your authentication service.

Other security considerations

The following is a list of factors that you should consider before running the production version of PAS for OpenEdge:

- SSL/TLS certificate
    - Add a valid key/certificate to your PASOE instance
    - Do NOT rely on the key/certificate sent by OpenEdge for testing

- For administration, add new users and assign PAS roles (ROLE_PSC**). If using a /WEB-INF/oeablSecurity-*-local.xml configuration modify the users.properties file.
- Modify the Tomcat manager users in ~/conf/tomcat-users.xml (only necessary if you add the Tomcat manager application).
- Modify the PAS for OpenEdge manager users ~/conf/tomcat-users.xml (only necessary if you add the PAS for OpenEdge REST API manager application).

Setting environment variables

If you want to set the same environment variables each time an instance is launched, create a *_setenv.sh or *_setenv.bat file and add it to the *instance_name*/bin directory.

When the instance is started, all files with the suffix _setenv.sh or _setenv.bat are run.

Note: Do not modify the existing openedge_setenv.sh or openedge_setenv.bat files. However, you can use them as templates to create your own _setenv files.