EMEA PUG 2024 OpenEdge OpenTelemetry Workshop

Your Workshop Environment

This repository will contain all of the resources required to be able to perform all tasks. We have Windows environments in the cloud setup for you to use, you will receive connection information from your workshop hosts.

If you want to perform this workshop on your own laptop, make sure to have at least installed:

- OpenEdge 12.8.3 (PASOE, Database, Client-Networking, PDSOE)
- Docker (Desktop)
- Git
- OpenEdge Command Center Server (optional)
- Postman (optional)
- cURL (optional)

> [!TIP] All installers and binaries required for this workshop can be found on the Workshop VM's in the *c:\workshop\binaries* folder. When using your own local machine you can either download these yourself or ask your workshop owners for a copy

APM Choices

There are quite some APM (Application Performance Monitoring) offerings available on the market, both open source and commercialy. We will use a centralized cloud instance of NewRelic to capture your progress in today's workshop and metrics from all of your PASOE instances.

Next to that, for your own excercises we decided to also use a combination of open source projects/products, all of them are at least included within the CNCF (Cloud Native Computing Foundation) to collect, prepare and visualize the metrics and tracing data.

Because setting up all of these products and their configurations is far outside of the scope of this workshop, we decided to use Docker (Compose) to compose all of these applications and have an easy way to stop and start all of those by a single command.

> [!NOTE] You can stop/start the whole 'OpenTelemetry stack' at any time with the following commands (executed on command line from the base directory of this workshop):
> **Stop**
>
> ```
> docker compose down -d
> ```
>
> **Start**
>
> ```
> docker compose up -d
> ```

The Docker Compose command will start a full environment that has configured running instances of:

- oTel Collector (the official OpenTelemetry Collector)
- Jaeger (Distributed Tracing System)
- Prometheus (Monitoring system and time series database)
- Grafana (Observability platform: query, visualize and alert on metrics)

Let's get it Started!

Tasks:

1. On your machine open CMD and navigate to your workshop directory ("c:\workshop")

```
cd c:\workshop
```

2. Clone this project into the workshop directory

```
git clone https://github.com/rwdroge/pug2024OTEL.git
```

OpenTelemetry Metrics

We will start this workshop by collecting Metrics in the OpenTelemetry standard for both a PASOE instance and a RDBMS instance. As mentioned earlier during the presentation, we can use an OpenEdge Command Center Agent to do this for us. We don't need an OpenEdge Command Center Server installation for this to work, but we have configured one for you nonetheless.

The OpenEdge Command Center Agent can be installed using a silent or interactive installer, but it can also be deployed using a set of configuration files (i.e. ideal for Docker deployments). Today we will use the 'normal' Windows installation method. We've already downloaded the latest version from ESD for you and put that into the binaries folder.

OpenEdge Command Center works with a Agent Key so that only verified Agents can connect to the Command Center Server. You can create and export an Agent key from the Command Center Console:

- Go to the 'Agent Keys' menu item
- Click the 'Generate Agent Keys' button to create a new Agent Key
- Either accept the default key name or change it and choose 'Save'
- In the next screen, choose 'Download Key File'
- Save the Agent Key file in the c:\workshop\

Now that we have our Agent key file that contains the secret and server configuration details, it's time to go ahead with the installation of the Command Center Agent!

Tasks:

1. Create and download an Agent Key file that you can use during the OpenEdge Command Center Agent installation
2. Start the OpenEdge Command Center Agent Installer (PROGRESS_OECC_AGENT_1.3.0_WIN_64.exe)
3. Click **Next** in the Introduction section
4. Accept the License Agreement and choose **Next**
5. Change the Java directory to: and leave other options as-is and choose **Next**
6. In the Server Connections section, select the earlier saved Agent Key file by clicking the **Choose..** button
7. Select the Agent Key file and choose **Open**

> [!NOTE] All other fields are automatically filled after selecting the Agent Key file

8. Choose **Next**
9. Select the OpenEdge Installation directory (DLC) and choose **Next**
10. Review the installation info and choose **Install**
11. Choose **Done**

> [!TIP] You can stop and start the OECC Agent as a Windows Service
> For troubleshooting, you can find the OECC Agent log files in *C:\Progress\OECC_Agent\logs*

OpenTelemetry Tracing

As discussed during the presentation, you can setup tracing for both ABL Clients and PASOE instances. For any ABL Client, we can add the -*otelConfig* parameter, followed by the file name that contains the OpenTelemetry configuration.

In the tracing/conf folder of this project, you will find a sample configuration file that can be refered to in either the .pf file for an ABL Client or as a new option *otelConfigFile* in the openedge.properties file in the [AppServer.SessMgr] sections for PASOE.

Open the Jaeger UI.

Notice that your traces in the 'Service' dropdown selection box is now showing '*empty-service-name*' for your traces.

Obviously this is not very useful as it will be hard to recognize where traces are coming from if they are all refering to the same 'empty-service-name'.

Luckily there is a way of setting additional properties using the OpenTelemetry config file in the form of 'resource attributes'.

By adding the following to the existing JSON configuration file for PASOE, we can send additional information to the OpenTelemetry Collector, which in turn shares that with Jaeger (and other exporters):

```
  "resource_attributes": "service.name=Demo,myservice=testopenedge,
location=PASOE",
```

> [!NOTE] This should be added to the OpenTelemetry Configuration and not the OpenEdge Telemetry configuration. You can add it above the exporters definition.

Of course, it makes sense to do something similar for the ABL client OpenTelemetry configuration as well, you will just change the 'Location' to reflect that this is not coming from PASOE but from a client.

Tasks:

1. Add the additional resource attributes to the OpenTelemetry configuration files for both PASOE and the ABL Client
2. Restart the ABL Client and PASOE instance
3. Rerun the requests from the ABL Client
4. Check Jaeger again and compare with before to verify your changes