Progress®



Unified Logging for PAS for OpenEdge Container

OpenEdge®

Copyright

© **2019 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.

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Deliver More Than Expected, Icenium, Kendo UI, Kinvey, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed 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, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, JustAssembly, JustDecompile, JustMock, Kinvey Chat, NativeChat, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, 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.

February 2019

Last updated with new content: Release 12.0

Updated: 2019/03/01

1

EFK Stack Logging for Containers

Each container has its own set of logs and sorting through all available logs in a filesystem is time consuming and inefficient. To alleviate this issue, Progress recommends using EFK (Elastic Fluentd Kibana) stack logging to create a unified logging structure. A unified logging structure allows you to access logs for every file across all of your containers in a single place. EFK stack logging uses Elasticsearch, Fluent Bit, and Kibana to collect and display log files from different containers.

For details, see the following topics:

- Implementation
- EFK Stack Logging in a Container
- PAS for OpenEdge log files

Implementation

The following figure illustrates the Progress recommended approach for unified logging:



1. Progress includes Fluent Bit within the PAS for OpenEdge containers which allows easy collection of logs. Fluent Bit is a log processor that collects logs for multiple sources and sends them to many different locations.

   Note: For more details, refer to the Fluent Bit documentation website.

2. In the second step Fluent Bit pushes the logs it collects to Elasticsearch.
3. Elasticsearch is a search engine that stores, searches, and analyzes huge volumes of data. In the EFK stack, Elasticsearch is used to read logs from Fluent Bit and analyze them for display in Kibana. Kibana is a web interface for visualizing the data collected from Elasticsearch.

Together these three functions make up the EFK stack.

The following components are required for using EFK stack logging with the PAS for OpenEdge container:

- A Docker environment
- The Docker Container for PAS for OpenEdge
- Fluent Bit
- Elasticsearch
- Kibana

EFK Stack Logging in a Container

Note: The code examples in these steps are taken from the pasoe-basic sample app. Fluent Bit is provided with the baseline PAS for OpenEdge container.

The following steps describe how to implement EFK stack logging in your PAS for OpenEdge container:

1. Ensure that Fluent Bit, Elasticsearch, and Kibana are all installed on your Docker host machine. If you are using these programs as Docker containers, you can use the `docker pull` command to bring them into your host environment.
2. Configure a `docker-compose.yaml` file. The following example is taken from the sample app. You can configure the environment variables based on your environment:

Note: The version numbers in the following example may not be the latest versions. If you are having issues check to make sure you have the latest version of Elasticsearch and Kibana.

```
version: '3'
services:
  web:
    image: nginx
    ports:
    - "8080:80"
    volumes:
    - ./webui:/usr/share/nginx/html:ro
  sports:
    build: .
    environment:
    - DBSERVER=<Place_IP_Address_of_DB>
    - EFK_LOGGING=true
    - INSTANCE_NAME=oepas1
    - APP_NAME=pasoe-basic-sample
    ports:
    - "8811:8811"
    stdin_open: true
    tty: true
    volumes:
    - ./conf/progress.cfg.pasoe:/psc/dlc/progress.cfg
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:6.4.2
    expose:
    - 9200
    ports:
    - "9200:9200"
  kibana:
    image: docker.elastic.co/kibana/kibana:6.4.2
    links:
    - "elasticsearch"
    ports:
    - "5601:5601"
```

Note: The configuration files for Fluent Bit point to a host called `elasticsearch` by default. This setting can be configured by system administrators to point to a different host. The following fields should be configured based on your environment:

- `DBSERVER` – Set this environment variable to the machine running your database.
- `EFK_LOGGING` – Set this environment variable to `true` to enable usage of the EFK Stack for log management with the Fluent Bit set up in the container.
- `INSTANCE_NAME` – Set this environment variable to the name of the PAS for OpenEdge instance.
- `APP_NAME` – Set this environment variable to the logical name for the application to display in Elasticsearch and Kibana.

---

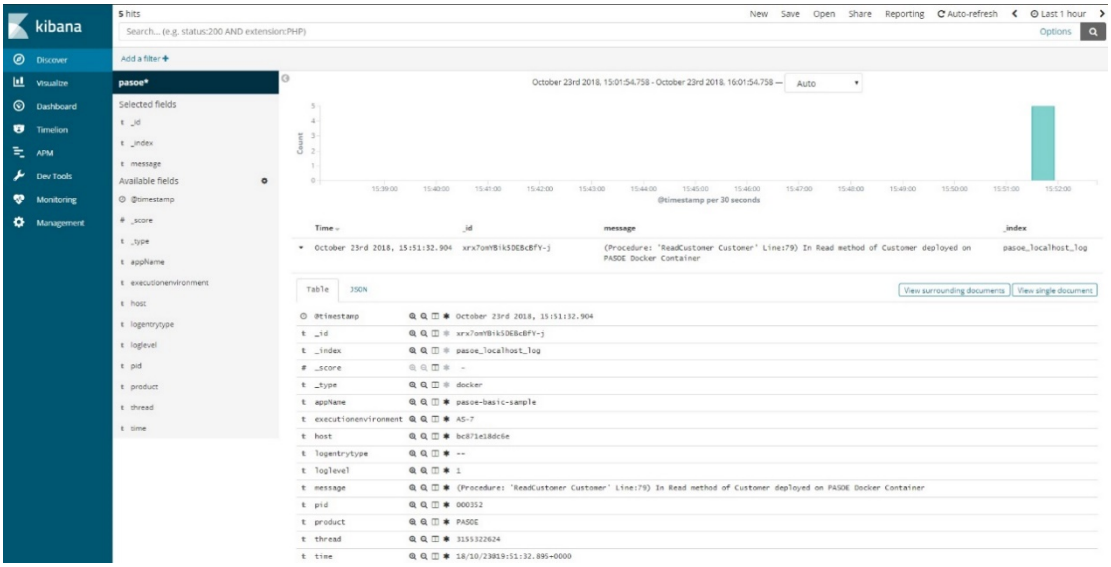
3. Build the Docker image:

```
docker-compose build
```

4. Run the Docker image, you should see that the Fluent Bit, Elasticsearch, and Kibana services started in your Docker host environment:

   ```
   docker-compose up -d
   ```

5. Access Kibana from a web browser to view the logs. The default URL is `http://<docker-host-machine>:5601`.
6. In the Management section of Kibana, create an index by specifying `pasoe_*`.
7. Click Discover to view your logs displayed in an index:



After this configuration is complete, any future entries made to the PAS for OpenEdge log files will be available in Kibana.

Note: By default Docker allocates 10 seconds to shutdown a container through the `docker-compose down` command. With PAS for OpenEdge, you want Docker to wait long enough for Fluent Bit to push all the shutdown logs of PAS for OpenEdge to Elasticsearch. To do so, use the following command:

```
docker-compose down -t <timeout_value>
```

PAS for OpenEdge log files

This topic lists all the default log files produced by PAS for OpenEdge. Use this list when setting up your implementation of EFK stack logging.

Note: Session manager, host manager, and manager logs require manager functionality and are not available in production PAS for OpenEdge by default.

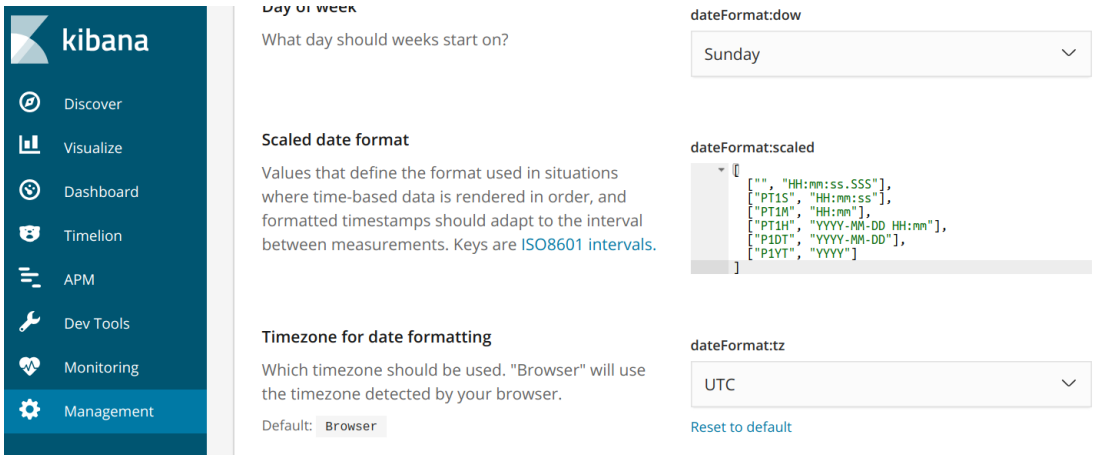
- Tomcat
  - `catalina.out`
  - `catalina.<date>.log`

  - `<localhost>.<date>.log`

- ABL Application
  - `<ABL-app-name>.<date>.log`
  - `<ABL-app-name>.agent.<date>.log`
- PASOE Authentication/Authorization Audit Trails (v 11.7.4+ & 12.0.0+)
  - `<ABL-app-name>_authn.<date>.log`
  - `<ABL-app-name>_authz.<date>.log`

Note: The two log files are not generated by default. They are created once the relevant events are triggered.

Date and time in the PAS for OpenEdge log files are recorded in the UTC +0 format. For their timestamps to display correctly in Kibana you will need to configure Kibana to be compatible with the PAS for OpenEdge logs. You can change the date formatting for Kibana in the Advanced Settings under the Management options:



Mapped Diagnostic Context (MDC) Support

The **Mapped Diagnostic Context (MDC)** support is included to allow the customer to add, remove, and rearrange fields to conform to the log-file and scanning standards of the enterprise, and to meet any enterprise policies.

The **Fluent BIT** configuration, which is a part of PAS for OpenEdge image, extracts metadata from the log message from various log files and enriches the log message sent to log processing systems. In release 12.0, wherever applicable, MDC fields are also extracted and added as metadata to the log message. For example:

```
{"@timestamp":"2019-01-16T12:08:04.880Z",
"date":"2019-01-16T12:08:04.880Z",
"instance":"oepas1",
"requestid":"Sports:f:00000002",
"eventstatus":"success",
"loginprincipal":"restuser",
"failreason":"",
"faildetails":"",
"hostaddress":"af41815a9a50",
"product":"PASOE",
"appName":"pasoe-basic-sample"}
```

In the above example, the MDC Request ID is extracted as well.

Note: For any modifications in the MDC Logging configuration, the Fluent Bit configurations should be updated.

For more details about MDC support and other details, see Configure logging in PAS for OpenEdge

Customize Fluent Bit configuration

Fluent Bit is a multi-platform log processor and forwarder in which data or logs from different sources are collected and forwarded to multiple destinations. It is fully compatible with Docker and Kubernetes environments.

Customize

EFK support includes a built-in set of Fluent Bit configuration files which can be overridden if required. The configuration files are located in `/etc/fluent-bit` inside the PAS for OpenEdge docker.

PAS for OpenEdge logs are generated in unstructured format. To structure those log files, Fluent Bit's **Parser** plugins are used. Parser plugins help to convert unstructured data into a structured one.

If the pattern of the PAS for OpenEdge logs are modified, the patterns in the Fluent Bit Parser files should be updated accordingly. The logs names and the patterns are self-explanatory. The Fluent Bit Parser file (`parser_pasoe_agent.conf`) is located `/etc/fluent-bit` inside the PAS for OpenEdge docker.

To update the configurations inside the docker image, follow either of the steps.

1. Copy the Fluent Bit configurations while creating the PAS for OpenEdge application docker image. For example: `COPY fluentbit/conf/* /etc/fluent-bit/.`
2. Use volume mount to replace the custom Fleunt Bit configurations. For example: `./fluent-bit/conf:/etc/fluent-bit`.

Log life cycle

Establishing a life cycle for your log files is important to minimize disk space usage in a container. You can set a daily rollover for log files in the `logging.xml` file in the `conf` directory using the `MaxHistory` option. Setting `MaxHistory` to 3 will keep log files for three days. The PAS for OpenEdge agent log file can use log rollover. To enable the feature, add the date to the agent log file in the `sports.props` file.

To delete the agent log file, system administrators can also use a cron job to run the following command daily to free up space.:

```
find logs -type f -name <"PASOE_LOG_NAME"> -mtime +3 -exec rm -vf {} \;
```

There are also instructions available in the Dockerfile for the sample application that describe how to use the `remove-old-jobs` script in the scripts directory.