# Table of Contents

1. Requirements .................................................................................................................. 1  
   1.1. Installation Requirements ......................................................................................... 1  
   1.2. Feature Requirements .............................................................................................. 1  
   1.3. Build (Install from Source) Requirements ................................................................. 1  

2. Get the APEX Source Code ............................................................................................... 2  

3. Build APEX ....................................................................................................................... 2  

4. Install APEX ...................................................................................................................... 4  
   4.1. Install with RPM and DPKG .................................................................................... 4  
   4.2. Install Manually from Archive (Unix, Cygwin) ....................................................... 5  
   4.3. Install Manually from Archive (Windows, 7Zip, GUI) ............................................. 5  
   4.4. Install Manually from Archive (Windows, 7Zip, CMD) ........................................... 6  

5. Build from Source ............................................................................................................ 7  
   5.1. Build and Install Manually (Unix, Windows, Cygwin) .............................................. 7  

6. Installation Layout ............................................................................................................ 9  

7. System Configuration ....................................................................................................... 9  
   7.1. APEX User and Group ............................................................................................. 9  
   7.2. Environment Settings: APEX_HOME and APEX_USER ......................................... 10  
   7.3. Edit the APEX Logging Settings ............................................................................. 11  
   7.4. Create Directories for Logging ............................................................................... 11  

8. Verify the APEX Installation ............................................................................................ 11  
   8.1. Verify Installation - run Engine ............................................................................... 12  
   8.2. Verify Installation - run an Example ..................................................................... 12  
   8.3. Verify a Full Installation - REST Editor ................................................................. 14  

9. Installing WAR Applications ............................................................................................ 16  

10. Running APEX in Docker ............................................................................................... 16  
    10.1. Run in ONAP ........................................................................................................ 16  
    10.2. Build a Docker Image ......................................................................................... 17
1. Requirements

APEX is 100% written in Java and runs on any platform that supports a JVM, e.g. Windows, Unix, Cygwin. Some APEX applications (such as the monitoring application) come as web archives, they do require a war-capable web server installed.

1.1. Installation Requirements

- Downloaded distribution: JAVA runtime environment (JRE, Java 8 or later, APEX is tested with the Oracle Java)
- Building from source: JAVA development kit (JDK, Java 8 or later, APEX is tested with the Oracle Java)
- A web archive capable webserver, for instance for the monitoring application
  - for instance Apache Tomcat
- Sufficient rights to install APEX on the system
- Installation tools depending on the installation method used:
  - ZIP to extract from a ZIP distribution
    - Windows for instance 7Zip
  - TAR and GZ to extract from that TAR.GZ distribution
    - Windows for instance 7Zip
  - RPM to install from the RPM distribution
    - Install: `sudo apt-get install rpm`
  - DPKG to install from the DEB distribution
    - Install: `sudo apt-get install dpkg`

1.2. Feature Requirements

APEX supports a number of features that require extra software being installed.

- Apache Kafka to connect APEX to a Kafka message bus
- Hazelcast to use distributed hash maps for context
- Infinispan for distributed context and persistence
- Docker to run APEX inside a Docker container

1.3. Build (Install from Source) Requirements

Installation from source requires a few development tools

- GIT to retrieve the source code
- Java SDK, Java version 8 or later
- Apache Maven 3 (the APEX build environment)
2. Get the APEX Source Code

The first APEX source code was hosted on Github in January 2018. By the end of 2018, APEX was added as a project in the ONAP Policy Framework, released later in the ONAP Casablanca release.

The APEX source code is hosted in ONAP as project APEX. The current stable version is in the master branch. Simply clone the master branch from ONAP using HTTPS.

```
git clone https://gerrit.onap.org/r/policy/apex-pdp
```

3. Build APEX

The examples in this document assume that the APEX source repositories are cloned to:

- Unix, Cygwin: `/usr/local/src/apex-pdp`
- Windows: `C:\dev\apex-pdp`
- Cygwin: `/cygdrive/c/dev/apex-pdp`

![A Build requires ONAP Nexus](image)
APEX has a dependency to ONAP parent projects. You might need to adjust your Maven M2 settings. The most current settings can be found in the ONAP oparent repo: [Settings](#).

![A Build needs Space](image)
Building APEX requires approximately 2-3 GB of hard disc space, 1 GB for the actual build with full distribution and 1-2 GB for the downloaded dependencies.

![A Build requires Internet (for first build)](image)
During the build, several (a lot) of Maven dependencies will be downloaded and stored in the configured local Maven repository. The first standard build (and any first specific build) requires Internet access to download those dependencies.

![Building RPM distributions](image)
RPM images are only build if the `rpm` package is installed (Unix). To install `rpm` run `sudo apt-get install rpm`, then build APEX.

Use Maven to for a standard build without any tests.

**Unix, Cygwin**

```
# cd /usr/local/src/apex-pdp
# mvn clean install -DskipTests
```

**Windows**

```
> cd \dev\apex-pdp
> mvn clean install -DskipTests
```

The build takes 2-3 minutes on a standard development laptop. It should run through without errors, but with a lot of messages from the build process.

When Maven is finished with the build, the final screen should look similar to this (omitting some `success` lines):
The build will have created all artifacts required for an APEX installation. The following example show how to change to the target directory and how it should look like.

Unix, Cygwin

```
# cd packages/apex-pdp-package-full/target
# ls -l
```

```
-rwxrwx---+ 1 esvevan Domain Users       772 Sep  3 11:55 apex-pdp-package-full_2.0.0~SNAPSHOT_all.changes*
-rwxrwx---+ 1 esvevan Domain Users 146328082 Sep  3 11:55 apex-pdp-package-full-2.0.0-SNAPSHOT.deb*
-rwxrwx---+ 1 esvevan Domain Users 15633 Sep  3 11:54 apex-pdp-package-full-2.0.0-SNAPSHOT.jar*
-rwxrwx---+ 1 esvevan Domain Users 146296819 Sep  3 11:55 apex-pdp-package-full-2.0.0-SNAPSHOT-
tarball.tar.gz*
drwxrwx---+ 1 esvevan Domain Users         0 Sep  3 11:54 archive-tmp/
-rwxrwx---+ 1 esvevan Domain Users        89 Sep  3 11:54 checkstyle-cachefile*
-rwxrwx---+ 1 esvevan Domain Users 10621 Sep  3 11:54 checkstyle-checker.xml*
-rwxrwx---+ 1 esvevan Domain Users 584 Sep  3 11:54 checkstyle-header.txt*
drwxrwx---+ 1 esvevan Domain Users        86 Sep  3 11:54 checkstyle-result.xml*
drwxrwx---+ 1 esvevan Domain Users        86 Sep  3 11:54 classes/
drwxrwx---+ 1 esvevan Domain Users        0 Sep  3 11:54 dependency-maven-plugin-markers/
drwxrwx---+ 1 esvevan Domain Users        0 Sep  3 11:54 etc/
drwxrwx---+ 1 esvevan Domain Users        0 Sep  3 11:54 examples/
drwxrwx---+ 1 esvevan Domain Users        0 Sep  3 11:55 install_hierarchy/
drwxrwx---+ 1 esvevan Domain Users        0 Sep  3 11:54 maven-archiver/
```

Windows

```
>cd packages\apex-pdp-package-full\target
>dir
```
4. Install APEX

APEX can be installed in different ways:

- Unix: automatically using `rpm` or `dpkg` from `.rpm` or `.deb` archive
- Windows, Unix, Cygwin: manually from a `.tar.gz` archive
- Windows, Unix, Cygwin: build from source using Maven, then install manually

4.1. Install with RPM and DPKG

The install distributions of APEX automatically install the system. The installation directory is `/opt/app/policy/apex-pdp`. Log files are located in `/var/log/onap/policy/apex-pdp`. The latest APEX version will be available as `/opt/app/policy/apex-pdp`. For the installation, a new user `apexuser` and a new group `apexuser` will be created. This user owns the installation directories and the log file location. The user is also used by the standard APEX start scripts to run APEX with this user's permissions.

RPM Installation

```
# sudo rpm -i apex-pdp-package-full-2.1.0-SNAPSHOT.rpm
                        preinst
arguments 1
                        postinst
arguments 1
```

creating group apexuser . . .
creating user apexuser . . .
DPKG Installation

```bash
# sudo dpkg -i apex-pdp-package-full-2.1.0-SNAPSHOT.deb
Selecting previously unselected package apex-uservice.
(Reading database ... 288458 files and directories currently installed.)
Preparing to unpack apex-pdp-package-full-2.1.0-SNAPSHOT.deb ...
********************preinst********************
arguments install
**********************************************
creating group apexuser . . .
creating user apexuser . . .
Unpacking apex-uservice (2.1.0-SNAPSHOT) ...
Setting up apex-uservice (2.1.0-SNAPSHOT) ...
********************postinst********************
arguments configure
******************************************************************************
```

Once the installation is finished, APEX is fully installed and ready to run.

### 4.2. Install Manually from Archive (Unix, Cygwin)

Download a tar.gz archive. Create a directory where APEX should be installed. Extract the tar archive. The following example shows how to install APEX in `/opt/apex` and create a link to `/opt/apex/apex` for the most recent installation.

```bash
# cd /opt
# mkdir apex
# cd apex
# mkdir apex-full-2.1.0-SNAPSHOT
# tar xvfz ~/Downloads/apex-pdp-package-full-2.1.0-SNAPSHOT.tar.gz -C apex-full-2.1.0-SNAPSHOT
# ln -s apex apex-pdp-package-full-2.1.0-SNAPSHOT
```

### 4.3. Install Manually from Archive (Windows, 7Zip, GUI)

Download a tar.gz archive and copy the file into the install folder (in this example C:\apex). Assuming you are using 7Zip, right click on the file and extract the tar archive. Note: the screenshots might show an older version than you have.
The right-click on the new created TAR file and extract the actual APEX distribution.

Inside the new APEX folder you see the main directories: bin, etc, examples, lib, and war

Once extracted, please rename the created folder to apex-full-2.1.0-SNAPSHOT. This will keep the directory name in line with the rest of this documentation.

4.4. Install Manually from Archive (Windows, 7Zip, CMD)

Download a tar.gz archive and copy the file into the install folder (in this example C:\apex). Start cmd, for instance typing Windows+R and then cmd in the dialog. Assuming 7Zip is installed in the standard folder, simply run the following commands (for APEX version 2.1.0-SNAPSHOT full distribution)

```
>c:
>cd \apex
>"\Program Files\7-Zip\7z.exe" x apex-pdp-package-full-2.1.0-SNAPSHOT.tar.gz -so | "\Program Files\7-Zip\7z.exe" x -aoa -si -ttar -o"apex-full-2.1.0-SNAPSHOT"
```
APEX is now installed in the folder `C:\apex\apex-full-2.1.0-SNAPSHOT`.

## 5. Build from Source

### 5.1. Build and Install Manually (Unix, Windows, Cygwin)

Clone the APEX GIT repositories into a directory. Go to that directory. Use Maven to build APEX (all details on building APEX from source can be found in APEX HowTo: Build). Install from the created artifacts (`rpm`, `deb`, `tar.gz`, or copying manually).

**Building RPM distributions**

RPM images are only build if the `rpm` package is installed (Unix). To install `rpm` run `sudo apt-get install rpm`, then build APEX.

The following example shows how to build the APEX system, without tests (`-DskipTests`) to save some time. It assumes that the APX GIT repositories are cloned to:

- **Unix, Cygwin**: `/usr/local/src/apex`
- **Windows**: `C:\dev\apex`

**Unix, Cygwin**

```
# cd /usr/local/src/apex
# mvn clean install -DskipTests
```

**Windows**

```
>c:
>cd \dev\apex
>mvn clean install -DskipTests
```

The build takes about 2 minutes without test and about 4-5 minutes with tests on a standard development laptop. It should run through without errors, but with a lot of messages from the build process. If build with tests (i.e. without `-DskipTests`), there will be error messages and stack trace prints from some tests. This is normal, as long as the build finishes successful.

When Maven is finished with the build, the final screen should look similar to this (omitting some success lines):

```
[INFO] tools .............................................. SUCCESS [  0.248 s]
[INFO] tools-common ..................................... SUCCESS [  0.784 s]
[INFO] simple-wsclient ................................... SUCCESS [  3.383 s]
[INFO] model-generator .................................... SUCCESS [  0.644 s]
[INFO] packages .......................................... SUCCESS [  0.336 s]
[INFO] apex-pdp-package-full .............................. SUCCESS [01:10 min]
[INFO] Policy APEX PDP - Docker build 2.0.0-SNAPSHOT ...... SUCCESS [ 10.307 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 03:43 min
[INFO] Finished at: 2018-09-03T11:56:01+01:00
[INFO] ------------------------------------------------------------------------
```

The build will have created all artifacts required for an APEX installation. The following example show how to change to the target directory and how it should look like.
Unix, Cygwin

```bash
# cd packages/apex-pdp-package-full/target
# ls -l
```

```
-rwxrwx---+ 1 esvevan Domain Users 772 Sep  3 11:55 apex-pdp-package-full_2.0.0~SNAPSHOT_all.changes*
-rwxrwx---+ 1 esvevan Domain Users 146328082 Sep  3 11:55 apex-pdp-package-full-2.0.0-SNAPSHOT.deb*
-rwxrwx---+ 1 esvevan Domain Users 15633 Sep  3 11:54 apex-pdp-package-full-2.0.0-SNAPSHOT.jar*
-rwxrwx---+ 1 esvevan Domain Users 146296819 Sep  3 11:55 apex-pdp-package-full-2.0.0-SNAPSHOT-tarball.tar.gz*
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 archive-tmp/
-rwxrwx---+ 1 esvevan Domain Users 89 Sep  3 11:54 checkstyle-cachefile*
-rwxrwx---+ 1 esvevan Domain Users 10621 Sep  3 11:54 checkstyle-checker.xml*
-rwxrwx---+ 1 esvevan Domain Users 584 Sep  3 11:54 checkstyle-header.txt*
-rwxrwx---+ 1 esvevan Domain Users 86 Sep  3 11:54 checkstyle-result.xml*
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 classes/
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 dependency-maven-plugin-markers/
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 etc/
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 examples/
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:55 install_hierarchy/
-drwxrwx---+ 1 esvevan Domain Users 0 Sep  3 11:54 maven-archiver/
```

Windows

```bash
>cd packages\apex-pdp-package-full\target
>dir
```

```
03/09/2018  11:55  <DIR>    .
03/09/2018  11:55  <DIR>    ..
03/09/2018  11:55  146,296,819  apex-pdp-package-full-2.0.0-SNAPSHOT-tarball.tar.gz
03/09/2018  11:55  146,328,082  apex-pdp-package-full-2.0.0-SNAPSHOT.deb
03/09/2018  11:54    15,633  apex-pdp-package-full-2.0.0-SNAPSHOT.jar
03/09/2018  11:55    772  apex-pdp-package-full_2.0.0~SNAPSHOT_all.changes
03/09/2018  11:54  <DIR>    archive-tmp
03/09/2018  11:54     89  checkstyle-cachefile
03/09/2018  11:54  10,621  checkstyle-checker.xml
03/09/2018  11:54    584  checkstyle-header.txt
03/09/2018  11:54    86  checkstyle-result.xml
03/09/2018  11:54  <DIR>    classes
03/09/2018  11:54  <DIR>    dependency-maven-plugin-markers
03/09/2018  11:54  <DIR>    etc
03/09/2018  11:54  <DIR>    examples
03/09/2018  11:55  <DIR>    install_hierarchy
03/09/2018  11:54  <DIR>    maven-archiver
8 File(s)  292,652,686 bytes
9 Dir(s)  14,138,720,256 bytes free
```

Now, take the .deb or the .tar.gz file and install APEX. Alternatively, copy the content of the folder `install_hierarchy` to your APEX directory.
6. Installation Layout

A full installation of APEX comes with the following layout.

$APEX_HOME
  └───war
      └───war

  ├───bin
  │   ├───binaries, mainly scripts (bash and bat) to start the APEX engine and applications
  │   └───etc
  │   │   └───configuration files, such as logback (logging) and third party library configurations
  │   └───examples
  │       ├───example policy models to get started
  │       │   ├───config
  │       │   │   └───configurations for the examples (with sub directories for individual examples)
  │       │   └───docker
  │       │       └───Docker files and additional Docker instructions for the exampples
  │       │   └───events
  │       │       └───example events for the examples (with sub directories for individual examples)
  │       │   └───html
  │       │       └───HTML files for some examples, e.g. the Decisionmaker example
  │       │   └───models
  │       │       └───the policy models, generated for each example (with sub directories for individual examples)
  │       │   └───scripts
  │       │       └───additional scripts for the examples (with sub directories for individual examples)
  │       │   └───lib
  │       │       └───the library folder with all Java JAR files
  │       │   └───applications
  │       │       └───applications, also known as jar with dependencies (or fat jars), individually deployable
  │       │   └───war
  │       │       └───WAR files for web applications

7. System Configuration

Once APEX is installed, a few configurations need to be done:

- Create an APEX user and an APEX group (optional, if not installed using RPM and DPKG)
- Create environment settings for $APEX_HOME and $APEX_USER, required by the start scripts
- Change settings of the logging framework (optional)
- Create directories for logging, required (execution might fail if directories do not exist or cannot be created)

7.1. APEX User and Group

On smaller installations and test systems, APEX can run as any user or group.

However, if APEX is installed in production, we strongly recommend you set up a dedicated user for running APEX.
This will isolate the execution of APEX to that user. We recommend you use the userid `apexuser` but you may use any user you choose.

The following example, for UNIX, creates a group called `apexuser`, an APEX user called `apexuser`, adds the group to the user, and changes ownership of the APEX installation to the user. Substitute `<apex-dir>` with the directory where APEX is installed.

```bash
# sudo groupadd apexuser
# sudo useradd -g apexuser apexuser
# sudo chown -R apexuser:apexuser <apex-dir>
```

For other operating systems please consult your manual or system administrator.

### 7.2. Environment Settings: APEX_HOME and APEX_USER

The provided start scripts for APEX require two environment variables being set:

- **APEX_USER** with the user under whose name and permission APEX should be started (Unix only)
- **APEX_HOME** with the directory where APEX is installed (Unix, Windows, Cygwin)

The first row in the following table shows how to set these environment variables temporary (assuming the user is `apexuser`). The second row shows how to verify the settings. The last row explains how to set those variables permanently.

<table>
<thead>
<tr>
<th>Unix, Cygwin (bash/tcsh)</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td># export APEX_USER=apexuser</td>
<td>&gt;set APEX_HOME=C:\apex\apex-full-2.1.0-SNAPSHOT</td>
</tr>
<tr>
<td># cd /opt/app/policy/apex-pdp/apex-pdp</td>
<td></td>
</tr>
<tr>
<td># export APEX_HOME=&quot;pwd&quot;</td>
<td></td>
</tr>
<tr>
<td># setenv APEX_USER apexuser</td>
<td></td>
</tr>
<tr>
<td># cd /opt/app/policy/apex-pdp/apex-pdp</td>
<td></td>
</tr>
<tr>
<td># setenv APEX_HOME &quot;pwd&quot;</td>
<td></td>
</tr>
<tr>
<td># env</td>
<td>grep APEX</td>
</tr>
<tr>
<td>APEX_USER=apexuser</td>
<td>APEX_HOME=/opt/app/policy/apex-pdp/apex-pdp</td>
</tr>
<tr>
<td>APEX_HOME=/opt/app/policy/apex-pdp/apex-pdp</td>
<td></td>
</tr>
</tbody>
</table>

7.2.1. Making Environment Settings Permanent (Unix, Cygwin)

For a per-user setting, edit the a user's `bash` or `tcsh` settings in `~/.bashrc` or `~/.tcshrc`. For system-wide settings, edit `/etc/profiles` (requires permissions).

7.2.2. Making Environment Settings Permanent (Windows)

On Windows 7 do

- Click on the **Start** Menu
- Right click on **Computer**
- Select **Properties**
On Windows 8/10 do

- Click on the **Start** Menu
- Select **System**

Then do the following

- Select **Advanced System Settings**
- On the **Advanced** tab, click the **Environment Variables** button
- Edit an existing variable, or create a new System variable: 'Variable name'="APEX_HOME", 'Variable value'="C:\apex\apex-full-2.1.0-SNAPSHOT"

For the settings to take effect, an application needs to be restarted (e.g. any open `cmd` window).

### 7.3. Edit the APEX Logging Settings

Configure the APEX logging settings to your requirements, for instance:

- change the directory where logs are written to, or
- change the log levels

Edit the file `$APEX_HOME/etc/logback.xml` for any required changes. To change the log directory change the line

```xml
<property name="VAR_LOG" value="/var/log/onap/policy/apex-pdp/" />
```

to

```xml
<property name="VAR_LOG" value="/PATH/TO/LOG/DIRECTORY/" />
```

On Windows, it is recommended to change the log directory to:

```xml
<property name="VAR_LOG" value="C:/apex/apex-full-2.1.0-SNAPSHOT/logs" />
```

Note: Be careful about when to use `\` vs. `/` as the path separator!

### 7.4. Create Directories for Logging

Make sure that the log directory exists. This is important when APEX was installed manually or when the log directory was changed in the settings (see above).

<table>
<thead>
<tr>
<th>Unix, Cygwin</th>
<th>Windows</th>
</tr>
</thead>
</table>
| `mkdir -p /var/log/onap/policy/apex-pdp`  
`chown -R apexuser:apexuser`  
`/var/log/onap/policy/apex-pdp` | `mkdir C:\apex\apex-full-2.1.0-SNAPSHOT\logs` |

### 8. Verify the APEX Installation

When APEX is installed and all settings are realized, the installation can be verified.
8.1. Verify Installation - run Engine

A simple verification of an APEX installation can be done by simply starting the APEX engine without any configuration. On Unix (or Cygwin) start the engine using `$APEX_HOME/bin/apexEngine.sh`. On Windows start the engine using `APEX_HOME\bin\apexEngine.bat`. The engine will fail to fully start. However, if the output looks similar to the following line, the APEX installation is realized.

```
Starting Apex service with parameters [...] ...
start of Apex service failed: Apex configuration file was not specified as an argument
org.onap.policy.apex.model.basicmodel.concepts.ApexException: Apex configuration file was not specified as an argument
  at org.onap.policy.apex.service.engine.main.ApexMain.<init>(ApexMain.java:68)
  at org.onap.policy.apex.service.engine.main.ApexMain.main(ApexMain.java:165)
usage: org.onap.policy.apex.service.engine.main.ApexMain [options...]
options
  -c,--config-file <CONFIG_FILE> the full path to the configuration file to use, the configuration file must be a Json file containing the Apex configuration parameters
  -h,--help outputs the usage of this command
  -m,--model-file <MODEL_FILE> the full path to the model file to use, if set it overrides the model file set in the configuration file
  -v,--version outputs the version of Apex
```

8.2. Verify Installation - run an Example

A full APEX installation comes with several examples. Here, we can fully verify the installation by running one of the examples.

We use the example called `SampleDomain` and configure the engine to use standard in and standard out for events. Run the engine with the provided configuration. Note: Cygwin executes scripts as Unix scripts but runs Java as a Windows application, thus the configuration file must be given as a Windows path.

```
# $APEX_HOME/bin/apexEngine.sh -c $APEX_HOME/examples/config/SampleDomain/Stdin2StdoutJsonEventJava.json
①
# $APEX_HOME/bin/apexEngine.sh -c C:/apex/apex-full-2.1.0-SNAPSHOT/examples/config/SampleDomain/Stdin2StdoutJsonEventJava.json
②
>&APEX_HOME\bin\apexEngine.bat -c %APEX_HOME\examples\config\SampleDomain\Stdin2StdoutJsonEventJava.json
③
```

① UNIX
② Cygwin
③ Windows

The engine should start successfully. Assuming the logging levels are not change (default level is `info`), the output should look similar to this (last few lines)
Starting Apex service with parameters [-c, v:/dev/ericsson/apex/onap/apex-pdp/packages/apex-pdp-package-full/target/install_hierarchy/examples/config/SampleDomain/Stdin2StdoutJsonEventJava.json] ...

2018-09-05 15:16:43,962 Apex [main] INFO o.o.p.a.s.e.e.EngDepMessagingService - engine<--deployment messaging starting ...
2018-09-05 15:16:43,963 Apex [main] INFO o.o.p.a.s.e.e.EngDepMessagingService - engine<--deployment messaging started
2018-09-05 15:16:45,120 Apex [main] INFO o.o.p.a.s.e.r.impl.EngineServiceImpl - Added the action listener to the engine

Started Apex service

Important are the last two line, stating that APEX has added the final action listener to the engine and that the engine is started.

The engine is configured to read events from standard input and write produced events to standard output. The policy model is a very simple policy.

The following table shows an input event in the left column and an output event in the right column. Past the input event into the console where APEX is running, and the output event should appear in the console. Pasting the input event multiple times will produce output events with different values.
8.3. Verify a Full Installation - REST Editor

APEX has a REST application for viewing policy models. The application can also be used to create new policy models close to the engine native policy language. Start the REST editor as follows.

```
# $APEX_HOME/bin/apexApps.sh rest-editor

> \$APEX_HOME\bin\apexApps.bat rest-editor
```

The script will start a simple web server (Grizzly) and deploy a war web archive in it. Once the editor is started, it will be available on localhost:18989. The last few line of the messages should be:

```
Apex Editor REST endpoint (ApexEditorMain: Config=[ApexEditorParameters: URI=http://localhost:18989/apexservices/, TTL=-1sec], State=RUNNING) started at http://localhost:18989/apexservices/
```

Now open a browser (Firefox, Chrome, Opera, Internet Explorer) and use the URL http://localhost:18989/. This will connect the browser to the started REST editor. The start screen should be as follows.
Now load a policy model by clicking the menu File and then Open. In the opened dialog, go to the directory where APEX is installed, then examples, models, SampleDomain, and there select the file SamplePolicyModelJAVA.json. This will load the policy model used to verify the policy engine (see above). Once loaded, the screen should look as follows.

Now you can use the REST editor. To finish this verification, simply terminate your browser (or the tab), and then use CTRL+C in the console where you started the REST editor.
9. Installing WAR Applications

APEX comes with a set of WAR files. These are complete applications that can be installed and run in an application server. All of these applications are realized as servlets. You can find the WAR applications in $APEX_HOME/war (UNIX, Cygwin) or %APEX_HOME%\war (Windows).

Installing and using the WAR applications requires a web server that can execute war web archives. We recommend to use Apache Tomcat, however other web servers can be used as well.

Install Apache Tomcat including the Manager App, see V9.0 Docs for details. Start the Tomcat service, or make sure that Tomcat is running.

There are multiple ways to install the APEX WAR applications:

- copy the .war file into the Tomcat webapps folder
- use the Tomcat Manager App to deploy via the web interface
- deploy using a REST call to Tomcat

For details on how to install war files please consult the Tomcat Documentation or the Manager App HOW-TO. Once you installed an APEX WAR application (and wait for sufficient time for Tomcat to finalize the installation), open the Manager App in Tomcat. You should see the APEX WAR application being installed and running.

In case of errors, examine the log files in the Tomcat log directory. In a conventional install, those log files are in the logs directory where Tomcat is installed.

The current APEX version provides the following WAR applications:

- client-deployment-2.1.0-SNAPSHOT.war - a client to deploy new policy models to a running engine
- client-editor-2.1.0-SNAPSHOT.war - the standard policy REST editor GUI
- client-monitoring-2.1.0-SNAPSHOT.war - a client for monitoring a running APEX engine
- client-full-2.1.0-SNAPSHOT.war - a full client with a one-stop-access to deployment, monitoring, and REST editor
- examples-servlet-2.1.0-SNAPSHOT.war - an example APEX servlet

10. Running APEX in Docker

Since APEX is in ONAP, we provide a full virtualization environment for the engine.

10.1. Run in ONAP

Running APEX from the ONAP docker repository only requires 2 commands:

1. Log into the ONAP docker repo

```
docker login -u docker -p docker nexus3.onap.org:10003
```

1. Run the APEX docker image

```
docker run -it --rm nexus3.onap.org:10003/onap/policy-apex-pdp:latest
```
10.2. Build a Docker Image

Alternatively, one can use the Dockerfile defined in the Docker package to build an image.

APEX Dockerfile

```bash
# Docker file to build an image that runs APEX on Java 8 in Ubuntu
FROM ubuntu:16.04

RUN apt-get update && \
    apt-get upgrade -y && \
    apt-get install -y zip unzip curl wget ssh iproute2 iputils-ping vim && \
    apt-get install -y software-properties-common && \
    add-apt-repository ppa:openjdk-r/ppa -y && \
    apt-get update && \
    apt-get install -y openjdk-8-jdk

# Create apex user and group
RUN groupadd apexuser
RUN useradd --create-home -g apexuser apexuser

# Add Apex-specific directories and set ownership as the Apex admin user
RUN mkdir -p /opt/app/policy/apex-pdp
RUN mkdir -p /var/log/onap/policy/apex-pdp
RUN chown -R apexuser:apexuser /var/log/onap/policy/apex-pdp

# Unpack the tarball
RUN mkdir /packages
COPY apex-pdp-package-full.tar.gz /packages
RUN tar xvfz /packages/apex-pdp-package-full.tar.gz --directory /opt/app/policy/apex-pdp
RUN rm /packages/apex-pdp-package-full.tar.gz

# Ensure everything has the correct permissions
RUN find /opt/app -type d -perm 755
RUN find /opt/app -type f -perm 644
RUN chmod a+x /opt/app/policy/apex-pdp/bin/*

# Copy examples to Apex user area
RUN cp -pr /opt/app/policy/apex-pdp/examples /home/apexuser

RUN apt-get clean

RUN chown -R apexuser:apexuser /home/apexuser/*

USER apexuser

ENV PATH /opt/app/policy/apex-pdp/bin:$PATH
WORKDIR /home/apexuser
```