

For information on how to install and enable the IBM Business Process Manager (BPM) Analytics Technology Demonstration, follow the steps in each of the sections in this document.

## Install Elasticsearch and Kibana

---

1. Install Elasticsearch by following the official guide at:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/installation.html>

2. Install Kibana by following the official guide at:

<https://www.elastic.co/guide/en/kibana/current/setup.html>

**Note:** The IBM BPM analytics might not support the 'current' version at the elastic.co, please choose the supported version guided by the README.md

## Enable the Dynamic Event Framework (DEF)

---

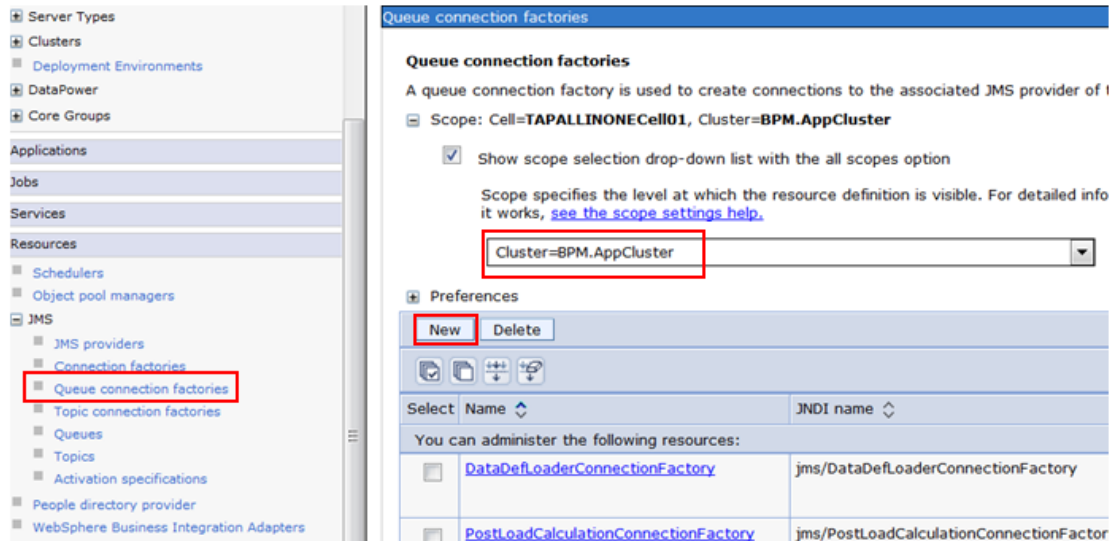
### Create WebSphere resources

Create a JMS queue connection factory

1. Create a JMS queue connection factory from the IBM Business Process Manager (BPM) administrative console, by clicking **Resources > JMS > Queue connection factories**.
2. Set the **Scope** to cluster, such as **Cluster=<Cluster\_Name>**.

**Note:** the queue connection factory is used by the DEF Emitter which belongs to the BPM engine, so the scope of the Queue connection factory is either the App cluster or cell.

3. Click **New** to create a new JMS queue connection factory:



4. Select a JMS resource provider, such as **Default messaging provider** and click **OK**:

[Queue connection factories](#) > **Select JMS resource provider**

Scope

Select the provider with which to create the Queue connection fa

☒ Default messaging provider

☐ WebSphere MQ messaging provider

OK Cancel

5. On the next configuration page, assign "**Name**" and "**JNDI name**", and select **Bus** for DEF usage:

## General Properties

### Administration

Scope

Cluster=BPM.AppCluster

Provider

Default messaging provider

\* Name

monitorcf

\* JNDI name

jms/monitorcf

Description

Category

### Connection

\* Bus name

BPM.BPM.Bus

Target

6. Select the authentication alias:

### Security settings

Select the authentication values for this resource.

Authentication alias for XA recovery

(none)

Mapping-configuration alias

(none)

Container-managed authentication alias

DeAdminAlias

Define a destination on the IBM BPM deployment environment bus

7. Define a destination on the IBM BPM deployment environment bus by clicking **Service integration > Buses** and select the bus. Click **Destinations** and add a new queue destination. The destination type is **Queue**:

Create new destination

Create a new destination on this bus.

Select destination type

☒ Queue

☐ Topic space

☐ Alias

☐ Foreign

Next Cancel

Assign a queue identifier:

→ Step 1: Set queue attributes

Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Set queue attributes

Configure the attributes of your new queue

\* Identifier  
MonitorDes

Description

Next Cancel

Select the bus member:

Create a new queue for point-to-point messaging.

Step 1: Set queue attributes

→ Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Assign the queue to a bus member

Assign the queue to a bus member that will st

Bus member  
Cluster=BPM.AppCluster

Previous Next Cancel

Create a JMS queue

1. Create a JMS queue by clicking **Resources > JMS > Queues**, select **Scope** to Cell, such as **Cell=<Cell\_Name>**.

**Note:** create the queue at the cell scope because the queue is shared by the DEF emitter as the write target and the BPMEmitter as the read source. For the three cluster topology, the BPMEmitter should be installed at the support cluster to avoid a resource conflict. Creating the queue at the cell scope can ensure that it is accessible by both the DEF emitter and the BPMEmitter.

2. Click **New** to create a new JMS queue, and select **Default messaging provider** as the JMS resource provider. On the configuration page, assign "**Name**" and "**JNDI name**", select Bus, and select the destination created in previous section as the Queue name:

**General Properties**

---

**Administration**

Scope

Provider

\* Name

\* JNDI name

Description

---

**Connection**

Bus name

\* Queue name

Delivery mode

Time to live  
 milliseconds

Priority

## Generate JSON native DEF event

There are two formats of DEF events - one is XML, the other is JSON. Normally, the performance of JSON native DEF events is better than XML DEF events. The use of JSON native DEF events improve overall performance.

BPM includes two sample scripts that you can use to configure the DEF to receive JSON native events. These scripts are in `<Install_Root>/BPM/Lombardi/tools/def`.

- SampleConfigureJSONEventsToJMS.py
- SampleReloadDEF.py

SampleConfigureJSONEventsToJMS.py is used to define the queue connection factory, queue, event subscriptions, and authentication alias to use.

SampleReloadDEF.py is used to make the DEF refresh its configuration dynamically.

Update the SampleConfigureJSONEventsToJMS.py script

1. Edit the sample script to update each of the fields **according to settings set in the previous sections**:

**defListenerId**: A string value that uniquely identifies this listener.

**eventQueueJndiName**: A string value that refers to the JNDI name of the queue resource created in WebSphere.

**eventQueueCFJndiName**: A string value that refers to the JNDI name of the queue connection factory resource created in WebSphere.

**eventQueueCF\_AuthorizationAlias**: A string value that refers to the authorization alias created in WebSphere.

For example:

```
# A unique id for the listener configuration to be created.
# This is a simple string that will be used to identify
# this DEF configuration.
defListenerId = 'jmsListenerForJSON1'

# The JNDI name of the target JMS queue. This is the JNDI name
# of an existing queue that has already been created on your
# deployment manager console.

eventQueueJndiName = 'jms/monQueue'

# The JNDI name of a connection factory appropriate for the target JMS queue
# identified by the eventQueueJndiName setting. This is the JNDI name
# of an existing queue that has already been created on your
# deployment manager console. The queue connection factory should be
# created and associated with BPM Server's bus. It should also specify
# the authorization alias which should match the value specified by
# the eventQueueCF_AuthorizationAlias parameter.

eventQueueCFJndiName = 'jms/monitorcf'

# Specify the Authentication Alias to use. This alias should
# be defined first on the deployment manager. In this sample, we are
# utilizing the deployment environment authorization alias that already
# exists.
eventQueueCF_AuthorizationAlias = 'DeAdminAlias'
```

2. Specify the subscription array. Each subscription in the subscriptions array is a single string with a '/' separator for each of the seven part keys. A comma is used to separate each subscription. The seven part keys are:

Application Name / Version / ComponentType / Component Name / Element Type /

Element Name / Nature

For a description of each of the parts, see the following topic:

[https://www.ibm.com/support/knowledgecenter/en/SS7NQD\\_8.5.7/com.ibm.wbpm.mon.imuc.doc/intro/intro\\_event\\_point\\_key.html](https://www.ibm.com/support/knowledgecenter/en/SS7NQD_8.5.7/com.ibm.wbpm.mon.imuc.doc/intro/intro_event_point_key.html)

To listen for every event for all applications, use the wildcard character as shown in the following example:

```
subscriptions=[  
'*/**/*/*/*/*/*/*/*'  
]
```

The following example shows how you might register to receive events for the Hiring Sample:

```
subscriptions=[  
'HSS/*/BPD/*/PROCESS/*/*',  
'HSS/*/BPD/*/ACTIVITY/*/*',  
'HSS/*/BPD/*/GATEWAY/*/*',  
'HSS/*/BPD/*/EVENT/*/*'  
]
```

3. Run the sample script from the command line. Go to the bin directory under your deployment manager profile home directory and run the SampleConfigureJSONEventsToJMS.py script. **Note:** Ensure that the support cluster (where DEF runs) is running before you run the script:

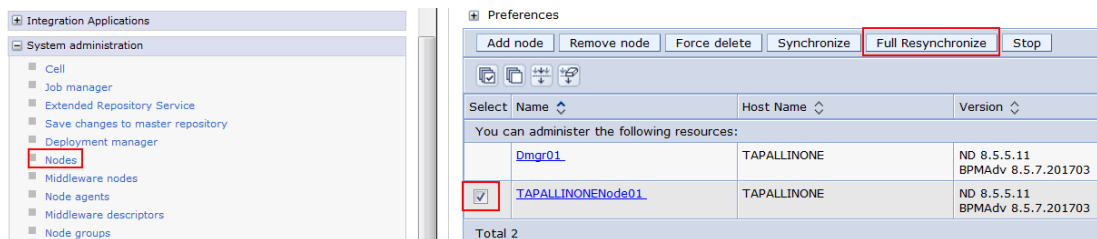
```
wsadmin -lang jython -f  
<Script_Location>/SampleConfigureJSONEventsToJMS.py
```

4. Run the SampleReloadDEF.py script to reload DEF. From a command line, go to the bin directory under your deployment manager profile home directory, run the following command:

```
wsadmin -lang jython -f <Script_Location>/SampleReloadDEF.py
```

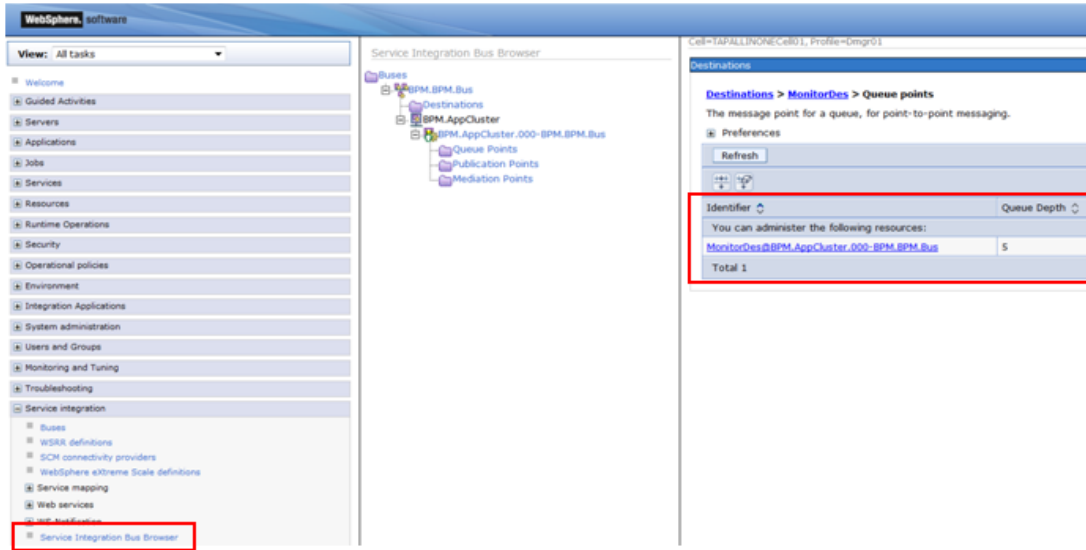
After running the sample script, a defconfig.xml file is created in the dmgr\_profile\_home/config/cells/cellName directory.

5. From the admin console, select **System administration > Nodes**, select nodes, and click **FullSynchronize**:



6. Restart Deployment Environment.

7. Validate that DEF was enabled successfully by starting a process in process portal. Select **Service Integration > Service Integration Bus Browser**, click into the bus destination which DEF used. The queue depth should be larger than 0:



(Not recommended) How to generate an XML DEF event

If you want to configure DEF to receive XML DEF events, update the SampleConfigureEventsToJMS.py script. It is also located at `<Install_Root>/BPM/Lombardi/tools/def`. The steps are similar to "Update the SampleConfigureJSONEventsToJMS.py script". Edit `defListenerId`, `eventQueueIndiName`, `eventQueueCFIndiName`, `eventQueueCF_AuthorizationAlias` and subscriptions fields. Do not reuse the same target JMS queue and connection factory for JSON DEF event generation.

After updating the SampleConfigureEventsToJMS.py script, run it from the command line. Go to the bin directory under your deployment manager profile home directory and run the SampleConfigureEventsToJMS.py script. **Note:** Ensure that the support cluster (where DEF runs) is running before you run the script:

```
wsadmin -lang jython -f <Script_Location>/SampleConfigureEventsToJMS.py
```

Run the SampleReloadDEF.py script to reload DEF. From a command line, go to the bin directory under your deployment manager profile home directory, run the following command:

```
wsadmin -lang jython -f <Script_Location>/SampleReloadDEF.py
```

After running the sample script, new `<defListener>` and `<defProducer>` should be added in `<Install_Root>/profiles/<Dmgr_Profile>/config/cells/<Cell_Name>/defconfig.xml`.

## How to disable Performance Data Warehouse (PDW)

If you want to disable PDW after enabling DEF, refer to

<https://developer.ibm.com/answers/questions/167196/disabling-tracking-data-generation-for-a-process-s.html>

## How to disable the Dynamic Event Framework (DEF)(optional)

If you want to disable DEF once you have finished with the Analytics function, follow these steps:



1. To delete the generated DEF configuration, go to  
 <Install\_Root>/profiles/<Dmgr\_Profile>/config/cells/<Cell\_Name>/defconfig.xml. Delete the following configurations:

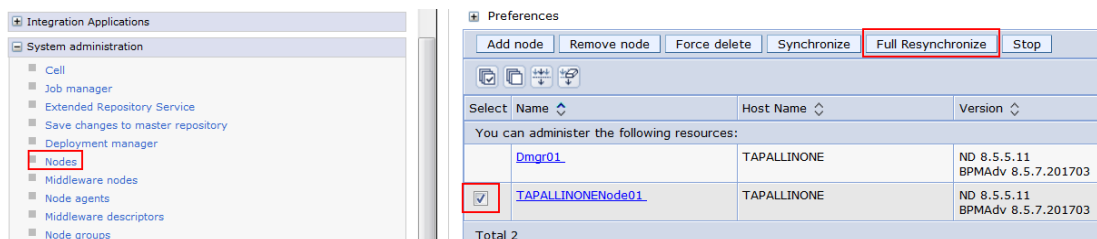
- defListener
- defProducer

```
<?xml version="1.0" encoding="UTF-8"?>
<xml:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:defconfig="http://www.ibm.com/websphere/appserv"
  <defconfig:DefListenerConfig xmi:id="DefListenerConfig_1490280302222">
    <defListener xmi:id="DefListener_1490280302277" listenerId="jmsListenerTHREE" listenerFactoryId="com.ibm.bpm.de
    <filter xmi:id="DefFilter_1490280302342" appName="*" version="*" componentType="*" componentName="*" elementT
    <defProperties xmi:id="DefProperty_1490280302301" name="JMS_QUEUE_JNDI" value="jms/monQueue"/>
    <defProperties xmi:id="DefProperty_1490280302314" name="JMS_QUEUE_CF_JNDI" value="jms/monitorcf"/>
    <defProperties xmi:id="DefProperty_1490280302327" name="JMS_AUTHENTICATION_ALIAS" value="DeAdminAlias"/>
    </defListener>
  </defconfig:DefListenerConfig>
  <defconfig:DefProducerConfig xmi:id="DefProducerConfig_1490280302405">
    <defProducer xmi:id="DefProducer_1490280302438" producerId="ProducerFor_jmsListenerTHREE">
    <filter xmi:id="DefFilter_1490280302459" appName="*" version="*" componentType="*" componentName="*" elementT
    </defProducer>
  </defconfig:DefProducerConfig>
</xml:XMI>
```

2. Run the SampleReloadDEF.py script to reload DEF. From a command line, go to the bin directory under your deployment manager profile home directory, run the following command:

```
Wsadmin -lang jython -f <Script_Location>/SampleReloadDEF.py
```

3. From the admin console, select **System administration >Nodes**, select nodes, click **Full Resynchronize**:



## How to generate task event (optional)

If you use REST API

([https://www.ibm.com/support/knowledgecenter/SSFTBX\\_8.5.7/com.ibm.wbpm.main.doc/topics/cdev\\_rest\\_apis.html](https://www.ibm.com/support/knowledgecenter/SSFTBX_8.5.7/com.ibm.wbpm.main.doc/topics/cdev_rest_apis.html)) or REST API test tool

([https://www.ibm.com/support/knowledgecenter/SSFTBX\\_8.5.7/com.ibm.wbpm.main.doc/topics/tdev\\_testingrestapis.html](https://www.ibm.com/support/knowledgecenter/SSFTBX_8.5.7/com.ibm.wbpm.main.doc/topics/tdev_testingrestapis.html)) or JavaScript API

([https://www.ibm.com/support/knowledgecenter/en/SSFPJS\\_8.5.7/com.ibm.wbpm.ref.doc/ae/doc/JSAPI.html](https://www.ibm.com/support/knowledgecenter/en/SSFPJS_8.5.7/com.ibm.wbpm.ref.doc/ae/doc/JSAPI.html)) to modify human tasks in a BPD process and you want DEF to generate task related messages, we

recommend that you enable the task related DEF message generation by adding the configuration below in 100custom.xml

([https://www.ibm.com/support/knowledgecenter/SSFTBX\\_8.5.7/com.ibm.wbpm.admin.doc/topics/managing\\_twks\\_config\\_settings.html](https://www.ibm.com/support/knowledgecenter/SSFTBX_8.5.7/com.ibm.wbpm.admin.doc/topics/managing_twks_config_settings.html)):

```
<common>
  <monitor-event-emission>
    <enable-task-api-def merge="replace">true</enable-task-api-def>
  </monitor-event-emission>
</common>
```

You can update the server configuration properties in the 100Custom.xml file to override the default configurations. The location of 100Custom.xml is specific to the server type you are configuring, where server type is performance-data-warehouse, process-center, or process-server. For example:  
*BPM\_INSTALL\_ROOT*/profiles/Dmgr01/config/cells/testServerCell01/nodes/testServerNode01/servers/BPM.AppCluster.testServerNode01.0/process-center/config/100Custom.xml.

Sync the node and restart the BPM server for the changes to take effect. You will see a TASK\_FIELD\_CHANGED message or TASK\_SUBJECT\_CHANGED message generated in the monitored queue if you modify human tasks:

```
<mon:monitorEvent mon:id="fd44650ea9c51792162189" >
  <mon:eventPointData>
    <mon:kind mon:version="2010-11-11" > wle:TASK_FIELD_CHANGED</mon:kind>
```

or

```
<mon:monitorEvent mon:id="x17ce6a4cfa9c51792162189" >
  <mon:eventPointData>
    <mon:kind mon:version="2010-11-11" > wle:TASK_SUBJECT_CHANGED</mon:kind>
```

## Skip KPI data to reduce performance downgrade (optional)

Enabling DEF will reduce the IBM BPM performance. Analysis shows that a sizable portion of downgrade is due to KPI data retrieval. If you do not need the KPI data to be added to DEF generated messages, add the following configuration in 100Custom.xml:

```
<common>
  <skip-kpi-data merge="replace">true</skip-kpi-data>
</common>
```

Skip KPI data will reduce performance downgrade. Location of 100Custom.xml is specific to the server type you are configuring, where server type is performance-data-warehouse, process-center, or process-server. For example:  
*BPM\_INSTALL\_ROOT*/profiles/Dmgr01/config/cells/testServerCell01/nodes/testServerNode01/servers/BPM.AppCluster.testServerNode01.0/process-center/config/100Custom.xml.

Sync the node and restart the BPM server for the changes to take effect.

# Configure and install the BPMEventEmitter

---

## Prerequisites and key words

Before configuring and installing the BPMEventEmitter file, you should:

- a. Enable the DEF message generation per the instruction on BPM server. At a minimum, you will need the **JMS monitor queue** and the **event queue JNDI name**.
- b. Know the target Elasticsearch server hosts. If security is enabled on the server, ensure that you have the **username**, **password** and **SSL/TLS related settings**.

## Update the configuration file

**Note:** the configuration file is in the WAR package, backup the configuration file after the modification is done, because any upgrade or reinstallation of the WAR package will cause the configuration file to be overwritten by the default.

1. Open the file **BPMEventEmitter.war** (provided as an artifact of this Technology Demonstration) with a zip tool, such as 7zip. Within the package, the configuration file is in BPMEventEmitter.war/WEB-INF/classes/config.yml

The configuration file is in YAML format:

```
# the configuration properties for the kafka if have
# the raw json events will be wrote to the kafka topic directly
# topic can be created automatically if it's not exist
# kafka bootstrap server(s) can be a list, separated by the comma
# by default, it's disabled
kafkaConfiguration:
kafka.bootstrap.servers: localhost:9092
monitor.topic: bpm-monitor-topic
enabled:false

# the configuration properties for the elastic search
# elastic search is the default event consumer
# the monitor event will be transformed to the query(kibana) optimized format
# before write to the ES as document
esConfiguration:
hosts: localhost:9200
enabled:true
# the following properties should be enabled when elastic search security is on
username: elastic
password:<xor>d3hrdXJEeXU=
httpsTrustType:
trustFileLocation:
hostnameVerifier:

# the identity for this BPM environment
# it can be the cell name or other proper identity
bpmCellName: bpmCell01

# the ES index name
esIndex: monitor
```

2. Update the values per your environment and save the updated file back to the war package. See below for more information about the configuration fields. **Note:** If you do not have Kafka installed on your environment, keep the default value and leave enabled set to “false”:

**bpmCellName:** Used as a field value of the generated message. If you want to analyze messages from different BPM clusters, update this field to different names across different BPM clusters.

**esIndex:** Used as the Elasticsearch index name where the generated data is stored.

**esConfiguration:** The configuration for Elasticsearch servers.

**Hosts:** Lists all server addresses in the Elasticsearch cluster. If you have not enabled security (SSL/TLS) on your Elasticsearch, use your IP directly. For example: 192.168.0.1:9200,127.0.0.1:9200

If you enabled security (SSL/TLS). You must input the https prefix to the address to announce that the HTTPS protocol should be used. For example: https://192.168.0.1:9200,https://127.0.0.1:9200

**username** and **password:** Used when you enable basic authentication on your Elasticsearch cluster. For the password field, you can use plain text or use an encoded password. You can use

**EventSummaryAgent** to encode your password by running the encodePassword function to get the encoded value as shown below:

```
EventSummaryAgent -encodePassword elastic
<xor>cXxraGFIdw==
```

If you enabled SSL/TLS on your Elasticsearch, you must set the **httpsTrustType** per the type you used. Three kinds are supported: ALL, CRT and JKS.

**ALL:** The application accepts all HTTPs communication. It should be used for test purposes only.

**CRT:** Provide the CA certificate file (.crt) to support that the application accepts certain HTTPs connections. In this option, you must provide the **trustFileLocation** setting with an absolute address. Ensure that the BPM server can access that file. For example:

```
httpsTrustType: CRT
trustFileLocation: /opt/IBM/BPM/elasticSearch.crt
```

**Default:** The agent accepts HTTPs communication accepted by the JVM default settings.

**hostnameVerifier:** Accepts Boolean values, leave as “false” in the production environment. If you are using the test environment which uses a CRT certificate including a wrong host or IP address, you can set that to true for testing purposes only.

### Security tips

Before enabling the Elasticsearch security, you can leave all of the security related fields empty or remove them from the configuration file to disable those settings.

After enabling the Elasticsearch security by installing X-Pack or protected that by using Nginx, you should provide the correct value for the fields.

For example:

```
hosts: https://<some_address>:9200
```

```
# the following properties should be enabled when elastic search security is on
username: elastic
password:<xor>cXxraGFIdw==
httpsTrustType: CRT
trustFileLocation: /opt/IBM/BPM/elasticsearch.crt
hostnameVerifier:false
```

This means that the Elasticsearch server protected with TLS and basic authentication. You have the CRT certification file located on /opt/IBM/BPM/elasticsearch.crt, and the CRT has the correct host/IP address to the server.

## Prepare the IBM BPM environment

The application will use JMS Activation specifications (AS) and Queues JNDI values to monitor the DEF events. Before installing the application, you must create the related JMS Activation specification first. For WebSphere Application Server Activation specifications and Queues, refer to the following topic:

[https://www.ibm.com/support/knowledgecenter/en/SSAW57\\_8.5.5/com.ibm.websphere.nd.doc/ae/SIBJMSActivationSpec\\_DetailForm.html](https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/SIBJMSActivationSpec_DetailForm.html)

The default AS JNDI and Queue JNDI used by our application will be:

AS: jms/defAS

Queue: jms/monQueue

You can create the activation specification and DEF monitor queue with that value. If you created the monitor queue or AS before with another name, you can update the value on file BPMEventEmitter.war/WEB-INF/ibm-ejb-jar-bnd.xml or update that value after the application is installed (see the next section of this document for more information).

For the default settings, assign the following when creating the AS:

**Name:** Any custom name

**JNDI name:** Use jms/ defAS

**Destination type:** Queue

**Destination JNDI name:** The Queue JNDI when you enable DEF on your server. Default will be jms/monQueue

**Bus name:** The bus name where the DEF message queue is located

**Security Settings > Authentication alias:** By default, Bus security is enabled on the BPM server. You must select the alias with Bus access, the default can be CellAdminAlias.

## Install the BPMEventEmitterWAR on the IBM BPM server

1. Log into the BPM server admin console, the default address is: <http://<Server Address>:9060/admin>

2. Expand **Applications > Application Types > WebSphere enterprise applications** in the left panel. In the right panel click **Install**.

3. **Select** the BPMEmitter.war file in local disk or input the remote address for that file, click **Next**.

4. **Select** Fast Path, click **Next**.

For "Step 1: Select installation options", if you do not have any special access control on your server, click **Next**.

For "Step 2: Map modules to servers", if you are using single cluster environment. Using default. If you are using golden topology with three cluster, you can map the application on support cluster only.

For "Step 3: Map context roots for Web modules", click **Next**.

For "Step 4: Metadata for modules", click **Next**.

For "Step 5: Summary", click **Finish**.

If the installation is successful, you will see a Save link as shown below:







To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:


- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

To work with installed applications, click the "Manage Applications" link.

5. After clicking Save, the application is installed:

	<a href="#">BPMEmitter.war</a>	
	<a href="#">BPMEmitter.war</a>	
	BSpaceEAR BPM.AppCluster	

6. There some **post actions** needed. Click the application name to open the configuration page. In the **Modules** section, click **Manage Modules**. Click BPMEmitter to open the module configuration page.

Select	Module	URI	Module Type	Server
	<a href="#">BPMEmitter</a>	BPMEmitter.war, WEB-INF/web.xml	Web Module	WebSphere: cell=Cell01, cluster=BPM.AppCluster

7. In the configuration page, update the Class loader order from Parent first to Parent last as shown below:

Configuration

General Properties

URI

BPMEventEmitter.war

Alternate deployment descriptor

Starting weight

10000

Class loader order

Classes loaded with local class loader first (parent last) ▼

Apply

OK

Reset

Cancel

Additional Properties

Stateful session bean failover settings

View Module Class Loader

Custom properties

Target specific application status

View EJB Deployment Descriptor

View Web Deployment Descriptor

Session Management

Web Module Proxy Configuration

8. Click **OK** and in the follow up page and click **Save** to save the change to the main WAS configuration.

**Optional:** If the AS and queue JNDI used is not the default value, and you have not updated them before installing the application, you can update that in the following page:

#### Enterprise Applications > BPMEventEmitter\_war > Message Driven Bean listener bindings

Select	Module	Bean	URI	Messaging type	Listener Bindings
<input type="checkbox"/>	BPMEventEmitter	BPMEventEmitterMDB	BPMEventEmitter.war,WEB-INF/ejb-jar.xml	javax.jms.MessageListener	<div> <div>Listener port</div> <div>Name</div> <div></div> </div> <div> <div>Activation Specification</div> <div>Target Resource JNDI Name</div> <div>jms/defAS</div> <div>Destination JNDI name</div> <div>jms/monQueue</div> <div>ActivationSpec</div> <div>authentication alias</div> <div></div> </div>

**Optional:** If you enabled Java 2 Security on your server, when starting the application you may receive an exception similar to the following:

```
SecurityManag W   SECJ0314W: Current Java 2 Security policy reported a potential
violation of Java 2 Security Permission. Refer to the InfoCenter for further
information.

Permission:

modifyThreadGroup : Access denied ("java.lang.RuntimePermission"
"modifyThreadGroup")
```

15

You must update the was.policy file to grant this application access. Further details on this are available from: [https://www.ibm.com/support/knowledgecenter/SSEQTP\\_8.5.5/com.ibm.websphere.base.doc/ae/tsec\\_was\\_policyfile.html](https://www.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/tsec_was_policyfile.html)

For this application, you must update the following file:

profile\_root/config/cells/cell\_name/applications/BPMEventEmitter\_war.ear/d  
eployments/BPMEventEmitter\_war/META-INF/was.policy

Update the contents as shown below:

```
//  
// Template policy file for enterprise application.  
// Extra permissions can be added if required by the enterprise  
// application.  
//  
// NOTE: Syntax errors in the policy files will cause the enterprise  
// application FAIL to start.  
//      Extreme care should be taken when editing these policy files.  
//      It is advised to use  
//      the policytool provided by the JDK for editing the policy files  
//      (WAS_HOME/java/jre/bin/policytool).  
//  
  
grant codeBase "file:${application}" {  
    permission java.lang.RuntimePermission "stopThread";  
    permission java.lang.RuntimePermission "modifyThread";  
    permission java.lang.RuntimePermission "modifyThreadGroup";  
    permission java.lang.RuntimePermission "createSecurityManager";  
};  
  
grant codeBase "file:${jars}" {  
};  
  
grant codeBase "file:${connectorComponent}" {  
};  
  
grant codeBase "file:${webComponent}" {
```



```
};

grant codeBase "file:${ejbComponent}" {
};
```

## Start the application

After the application is installed, it can process the messages. The application will retrieve message from the DEF monitor event queue, transform them into JSON objects, and target the Elasticsearch server.

1. Start the application on the admin console page. The default address is:

<http://<Server Address>:9060/admin>

2. After logging in, expand **Applications > Application Types > WebSphere enterprise applications** in the left panel. In the right panel, select BPMEmitter\_war then click **Start**.

After a successful start, you may see version and other information in SystemOut.log similar to the following:

```
[5/18/17 12:15:16:466 IST] 00000152 LifeCycleMana I Application Name:IBM BPM
Analytics
[5/18/17 12:15:16:466 IST] 00000152 LifeCycleMana I Application
Version:8.5.7.201703
[5/18/17 12:15:16:467 IST] 00000152 LifeCycleMana I Build Level:20170518_28
[5/18/17 12:15:49:509 IST] 0000b44b LifeCycleMana I I: DEF2ES MDB
started.
[5/18/17 12:15:49:509 IST] 0000b44b ConfigConnect I I: Kafka connection
disabled.
[5/18/17 12:15:49:727 IST] 0000b44b ConfigConnect I I: ElasticSearch
connection created.
```

For any issues, check the Troubleshooting section at the end of this document.

## Install, configure and run the EventSummaryAgent

The EventSummaryAgent is an agent application which will monitor the new events added in the configured Elasticsearch index and then generate summary events.

### Prerequisites and key words

Before configuring the EventSummaryAgent, you must:

- Know the target Elasticsearch server hosts. If security is enabled on server, know the **username**, **password** and **SSL/TLS related settings**.
- Know the Elasticsearch **index** used to store the data generated by BPMEmitter.

- c. Have a working JVM installed and correct JAVA\_HOME set.

## Install and update the configuration file

In the package, find the **EventSummaryAgent.tar** file and save it to the server where you will run this agent. You can find the sample configuration file in `EventSummaryAgent/conf/EventSummaryAgent.yml`

The configuration file is in YAML format:

```
#Facade type: elasticSearch
facadeType: elasticSearch

# Time gap used by event picker in milliseconds
# Only pickup events which is older than the value of pickerTimeGap to make sure
eventually sequencing
pickerTimeGap: 60000

# Whether enable the business data aggregator
enableBusinessDataAggregator: true

# Quality of service. Provided two mode now: strictMode | ignoreError
# strickedMode - will retry when meet error to prevent data lost on summary types.
# ignoreError - Will ignore error and continue processing. The summary type may not
complete when error happens.
qualityOfService: strickedMode

# the configuration properties for the source elastic search
# for the source event, the EventSummaryAgent only read the data without update
operation
# that can make sure it's safe to use ES index alias with multiple indices
configured
esSourceConfiguration:
  hosts: localhost:9200
  index: monitor
  # the following properties should be enabled when elastic search security is on
  username: elastic
  password: <xor>d3hrdXJEeXU=
  httpsTrustType:
  trustFileLocation:
  hostnameVerifier:
# the configuration properties for the target elastic search
# write the combined summary event as the target
# Notes: the index alias with multiple indices do not support write operation
esTargetConfiguration:
  hosts: localhost:9200
  index: monitor
  # the following properties should be enabled when elastic search security is on
  username: elastic
  password: <xor>d3hrdXJEeXU=
  httpsTrustType:
  trustFileLocation:
  hostnameVerifier:

# the progress will be recorded at the target Elasticsearch
progressStorage:
  # Index name used to store progress data
```

```

esIndex: oiprogress
# Type name used to store progress data
esType: readcursor
# Index name used to store restore task
esTaskIndex: restore_task_index

# archive related configurations
archive:
  enabled: false
# In object store we will merge several raw events into one object file
# minBulkSize controls the min events count for a bulk when possible
# maxBulkSize controls the max events count for a bulk.
# You should adjust that according to the event average size. It cannot bigger than
10000
  minBulkSize: 100
  maxBulkSize: 1000
# the configuration properties for the object storage
# we support object storage which use openstack-swift APIs
# Because different authentication method may used by openstack-swift
# So different setting sets may used
objectStoreConfiguration:
  # Set 1. Sample for using openstack swift interface with tempauth
  #=====
  providerOrApi: swift-tempAuth
  identity: test:tester
  credential: testing
  endpoint: http://localhost:8080/auth/v1.0
  #You can assign the container name where store the files.
  #If no the default one will be used
  containerName: BPMAntalyticArchive
  #swift need location to be settled. If no the first location we find will be
used
  #location: reginOne
  #If you are using private object storage without valid certification for HTTPs.
You should
  #enable below settings:
  #overrides:
  #   jclouds.relax-hostname: true
  #   jclouds.trust-all-certs: true
  #=====
  #Set 2. Sample for using openstack swift v3 auth (IBM Bluemix object storage
supported)
  #=====
  #providerOrApi: swift-v3
  #containerName: BPMAntalyticArchive
  #userId: 3eef42de3e3*****
  #password: L21i*****
  #auth_url: https://identity.open.softlayer.com
  #domainName: 136****
  #project: object_storage_1c8707e5_2480_****_b6ca_*****
  #If you are using private cloud without valid SSL certification for HTTPs. Set
below parameter to true.
  #disableSSLVerification: true
  #=====

```

```

# Merge the tracked fields cross different tracking groups.
# If you use the sub process or linked process, the tracking fields at the main
# process and the sub/linked
# process will be stored in the different tracking groups, but logically these
# tracked fields in the different
# tracking groups might stand for the same business entity.
# At this situation, you might need correlate the tracked fields which belong to
# the same process instance
# and in the different tracking groups together.
#
# As the example configuration show, you can configure the correlation field name
# per application and per BPD process.
# If you want set an global correlation field name, you can put a wildcard (*) as
# the processApplicationName
# and the processName
# NOTE: 1. assign multiple correlation fields in the same process and same
# application is not supported
#       2. cross process instance correlation is not supported
#       3. use the field name as the correlation, the value of the fields will be
# decided by the latest generated
#       activity event
#       4. the processName is the BPD name at the main process
#
# innerProcessCorrelation:
#   - processApplicationName: Application_Full_Name_1
#     processName: Process_Name_1
#     correlationFieldName: Tracked_FieldName_1
#   - processApplicationName: Application_Full_Name_2
#     processName: Process_Name_2
#     correlationFieldName: Tracked_FieldName_2

```

Update the values per your environment, you can update directly or save a copy and update:

**facadeType:** Keep the default value.

**esDataIndex:** Used as the Elasticsearch index name where it gets the events generated by BPMEventEmitter and stores the generated data.

**pickerTimGap:** A time gap used internal to prevent random data order. The suggested time is at least 10 seconds. The value is in milliseconds.

**enableBusinessDataAggregator:** Enable the default business data aggregator, it can aggregate the tracking groups by the same starting process instance. The default value is true. For the detailed description, see the section – Handling the Tracked Business Data of this document

**qualityOfService:** Quality of service which is used to control the agent behavior when errors are received. There are two available modes:

strickedMode - will retry when an error is received to prevent data lost on summary types.

ignoreError - will ignore the error and continue processing. The summary type may not complete when an error occurs.

**esSourceConfiguration:** The configurations for Elasticsearch raw event storage.

**esTargetConfiguration:** The configurations for Elasticsearch combined event storage.

**Hosts:** Lists all of the server addresses in the Elasticsearch cluster. If you have not enabled security (SSL/TLS) on your Elasticsearch, you can use the IP directly. For example:

192.168.0.1:9200,127.0.0.1:9200

If you enabled security (SSL/TLS), you must input an “https” prefix to the address to announce that the communicate should use HTTPS protocol. For example:

https://192.168.0.1:9200,https://127.0.0.1:9200

**username** and **password:** Used when you enabled basic authentication on your Elasticsearch cluster. For password field, you can use plain text or use an encoded password. To create an encoded password you can use the **EventSummaryAgent** with encodePassword parameter, as shown below:

```
EventSummaryAgent.sh/bat -encodePassword elastic  
<xor>cXxraGFIdw==
```

If you enabled SSL/TLS on your Elasticsearch, you must set the **httpsTrustType** per the type you used. Three kinds are supported: ALL, CRT and JKS.

**ALL:** The application accepts all HTTPs communication. It should be used for test purposes only.

**CRT:** Provide the CA certificate file (.crt) to support that the application accepts certain HTTPs connections. In this option, you must provide the **trustFileLocation** setting with an absolute address. Ensure that the BPM server can access that file. For example:

```
httpsTrustType: CRT  
trustFileLocation: /opt/IBM/BPM/elasticSearch.crt
```

**Default:** The agent accepts HTTPs communication accepted by the JVM default settings.

**hostnameVerifier:** Accepts Boolean values, leave as “false” in the production environment. If you are using the test environment which uses a CRT certificate including a wrong host or IP address, you can set that to true for testing purposes only.

**archive:** This configuration is used for archive function related configurations. The archive function can save your data into object storage during processing. And you can restore the data when they are lost in elasticsearch. Please refer to Data archive and restore section.

## Security tips

Before enabling the Elasticsearch security, you can leave all the security related fields empty or remove them from the configuration file to disable those settings.

After enabling the Elasticsearch security by installing X-Pack or protected that by using Nginx, you should provide the correct value for the fields.

For example:

```
hosts: https://<some_address>:9200  
# the following properties should be enabled when elastic search security is on
```

```
username: elastic
password:<xor>cXxraGFIdw==
httpsTrustType: CRT
trustFileLocation: /opt/IBM/BPM/elasticsearch.crt
hostnameVerifier:false
```

This means that the Elasticsearch server is protected with TLS and basic authentication. You have the CRT certification file located on /opt/IBM/BPM/elasticsearch.crt, and the CRT has the correct host/IP address to the server.

## Command line parameters

Usage of the EventSummaryAgent:

```
EventSummaryAgent.sh/bat [-configFile <configFileLocation>] | -encodePassword
<passwordValue> | -version | -restore | -help
```

**-configFile<configFileLocation>**: The optional -configFile option is the configuration file location you created. You can use this parameter when you start the agent. If you do not specify the - configFile option, the default configuration file is located at <root\_folder>/conf/ EventSummaryAgent.yml

**-encodePassword<passwordValue>**: Use this parameter to generate an encoded password as indicated in the configuration.

**-version**: Will print version information.

**-restore**: This option will start the event summary agent with restore mode. For detail please read archive and restore section.

**-help**: Will print help message.

For any issues, check the troubleshooting section at the end of this document.

## Start the EventSummaryAgent

To start the EventSummaryAgent, run the following command at the utility's <root\_folder>/bin:

For Linux:

```
./EventSummaryAgent.sh
or
./EventSummaryAgent.sh -configFile ../conf/EventSummaryAgent.yml
```

For Windows:

```
EventSummaryAgent.bat
or
EventSummaryAgent.bat -configFile ../conf/EventSummaryAgent.yml
```

If the agent starts successfully, you may find a log entry similar to the following:

```
<time stamp>com.ibm.bpm.mon.oi.combine.EventSummaryAgentdoJob  
INFO: I: Start getting process/activity events from the data source.
```

This message indicates that the agent is ready to begin processing the messages.

To stop the utility, send the kill signal using Ctrl+C from the command window, it will stop the EventSummaryAgent.

When running the EventSummayAgent in the background, use the “kill -2 <process\_id>” to stop it.

## Import the dashboard definition

### Prerequisites and key words

Unzip BPMDashboardKibana.zip, you will get two JSON files which are used to define dashboards in Kibana:

- BPMDefaultDashboard.json
- HiringSampleDashboard.json

Before importing the dashboard definition **BPMDefaultDashboard.json** to Kibana, you should:

1. Ensure **BPMEventEmitter.war** is started
2. Ensure **EventSummaryAgent** is started

**Note:** You should have one process instance with a user task completed before importing the dashboard definition, otherwise you may receive the following error due to a Kibana limitation (<https://github.com/elastic/elasticsearch/issues/22438>):

34 ⚠ Saved "field" parameter is now invalid. Please select a new field. OK 111s

### Create an index alias (Optional)

If you are not using **monitor** as the index name, you need to map all the indexes you used to **monitor** alias. For the detail API, refer to <https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-aliases.html>

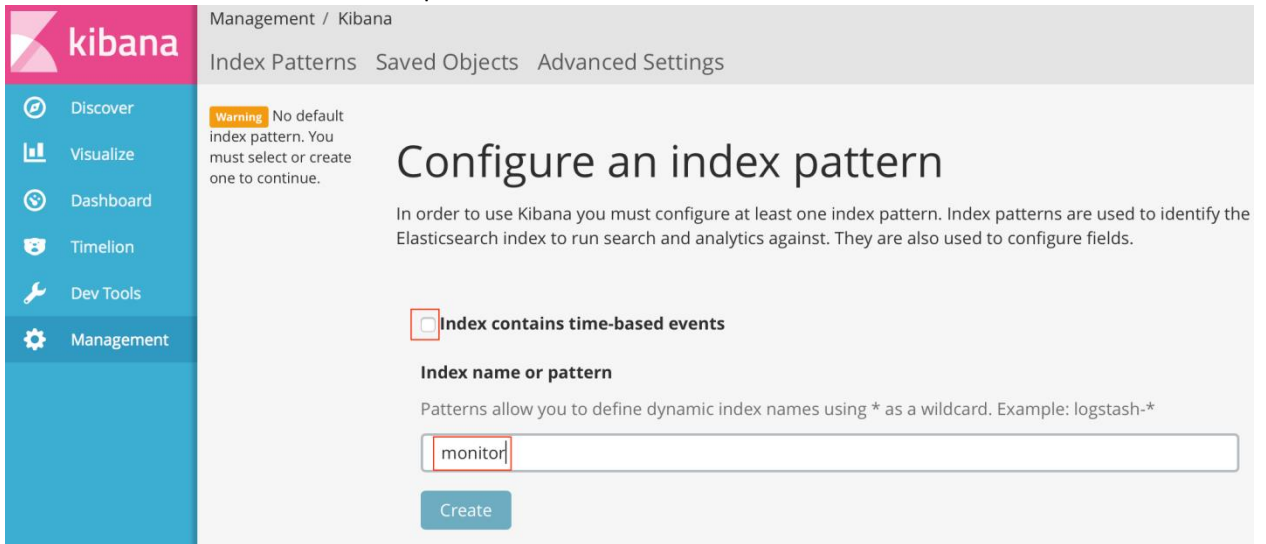
The following example is using a REST API in Kibana Dev Tools to set **monitor** as alias of index **monitor2**:

```
POST /_aliases  
{  
  "actions": [  
    { "add": { "index": "monitor2", "alias": "monitor" } }  
  ]  
}
```

### Create Index Patterns

1. In Kibana, click **Management > Index Patterns**, uncheck **Index contains time-based events**.

2. Enter **monitor** as the **Index name or pattern**, click **Create**:



Management / Kibana

Index Patterns Saved Objects Advanced Settings

**Warning** No default index pattern. You must select or create one to continue.

## Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

☐ Index contains time-based events

**Index name or pattern**

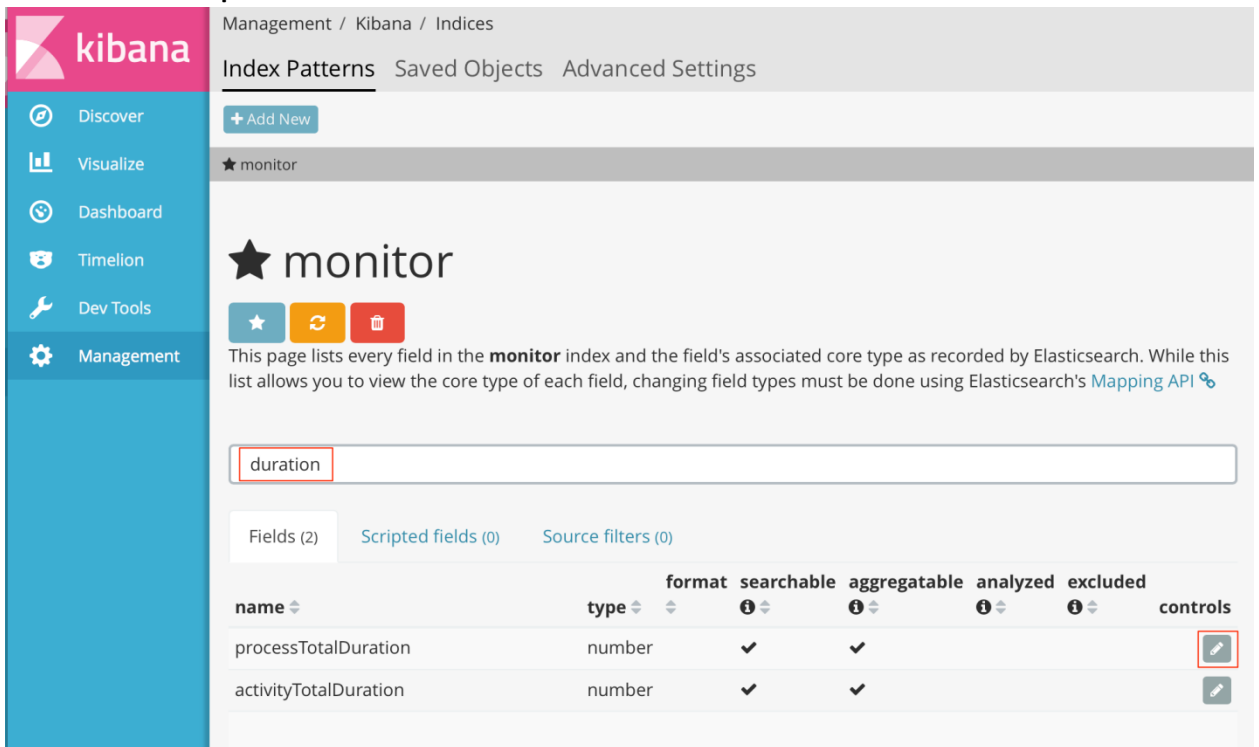
Patterns allow you to define dynamic index names using \* as a wildcard. Example: logstash-\*

monitor

Create

3. Search **Duration**, you will see two fields: **processTotalDuration** and **activityTotalDuration**

4. Click  beside **processTotalDuration**



Management / Kibana / Indices

Index Patterns Saved Objects Advanced Settings

+ Add New



★ monitor

## ★ monitor

This page lists every field in the **monitor** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's [Mapping API](#).

duration

Fields (2) Scripted fields (0) Source filters (0)

name	type	format	searchable	aggregatable	analyzed	excluded	controls
processTotalDuration	number		✓	✓			
activityTotalDuration	number		✓	✓			



5. Select **Duration** for Format, **Milliseconds** for Input Format, **Minutes** for Output Format. This will help to transform processTotalDuration from Milliseconds to Minutes.

The screenshot shows the Kibana interface with the 'Management' sidebar. The main content area is titled 'monitor' and 'processTotalDuration'. Under the 'Type' section, 'number' is selected. In the 'Format' section, 'Duration' is selected. Below this, the 'Input Format' is set to 'Milliseconds' and the 'Output Format' is set to 'Minutes'. The 'Decimal Places' are set to '2'. A warning icon is visible next to the 'Format' dropdown.

6. Repeat step 4 to step 5 to change the format for activityTotalDuration.

## Import the dashboard definition

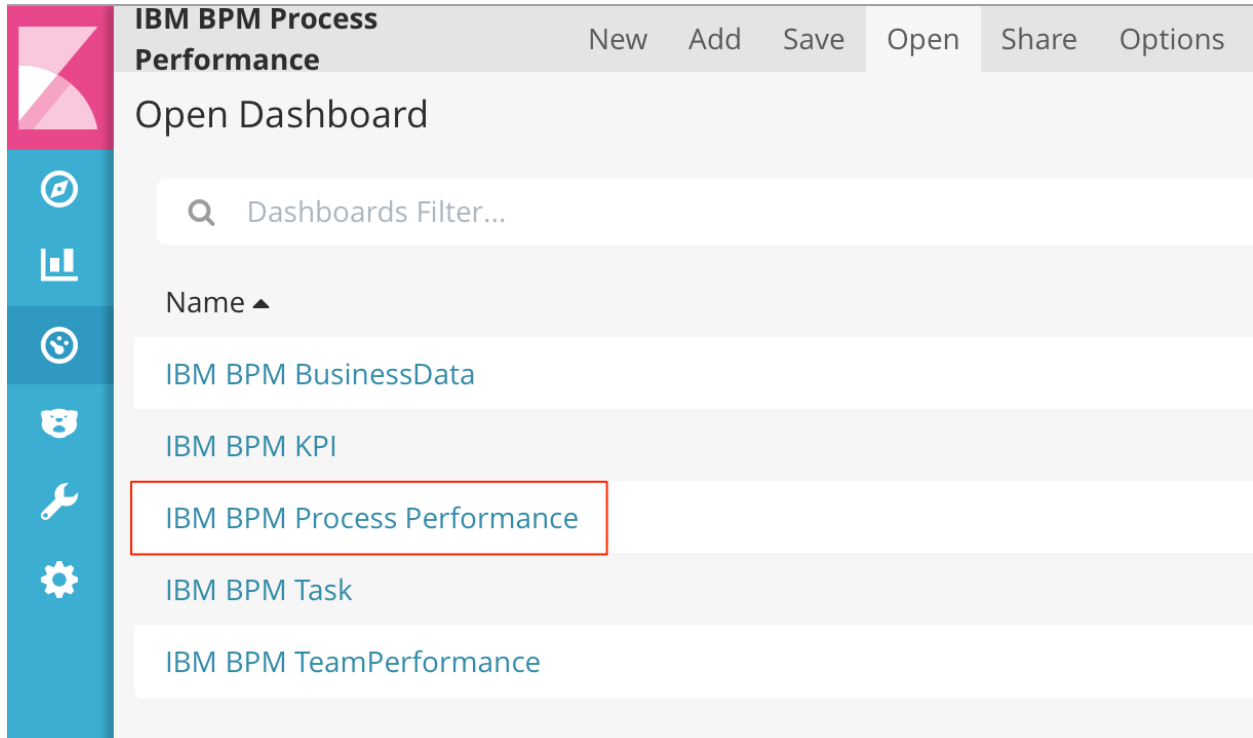
1. In Kibana, select **Management > Saved Objects**.
2. Click **Import**, select **BPMDefaultDashboard.json**.
3. After importing, you can see **5** extra Dashboards, **6** extra Searches, and **24** extra Visualizations:

The screenshot shows the Kibana 'Edit Saved Objects' page. The 'Import' button is highlighted with a red box. Below the buttons, there is a filter input and three tabs: 'Dashboards (5)', 'Searches (6)', and 'Visualizations (24)'. A list of objects is shown with checkboxes and buttons for 'Delete' and 'Export'. The objects listed are:

- IBM BPM BusinessData
- IBM BPM KPI
- IBM BPM Process Performance
- IBM BPM Task
- IBM BPM TeamPerformance

## Open the imported dashboard

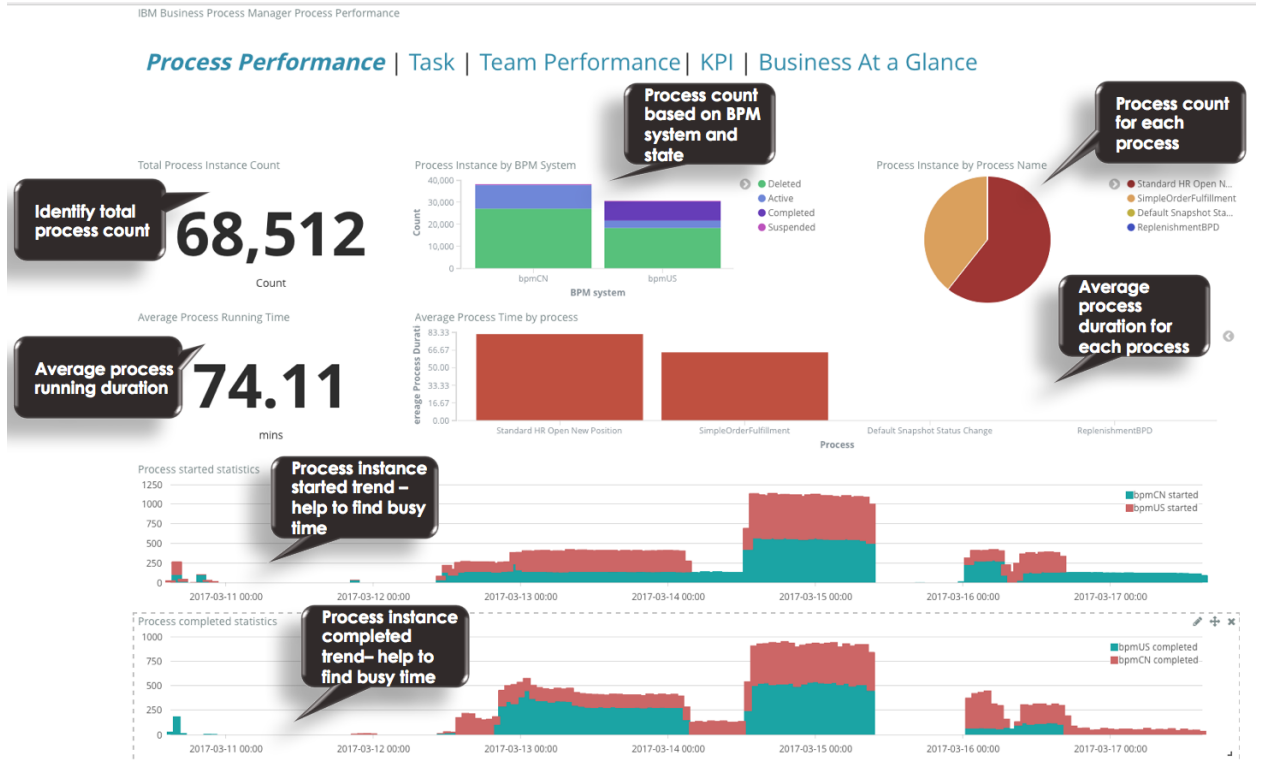
In Kibana, select **Dashboard**, open **IBM BPM Process Performance**:



The screenshot shows the Kibana dashboard interface. At the top, there's a header for 'IBM BPM Process Performance' with buttons for 'New', 'Add', 'Save', 'Open', 'Share', and 'Options'. Below this is a section titled 'Open Dashboard'. On the left, there's a sidebar with icons for various Kibana features. The main area contains a search bar labeled 'Dashboards Filter...' and a list of dashboards under the heading 'Name ▲'. The list includes 'IBM BPM BusinessData', 'IBM BPM KPI', 'IBM BPM Process Performance' (which is highlighted with a red box), 'IBM BPM Task', and 'IBM BPM TeamPerformance'.

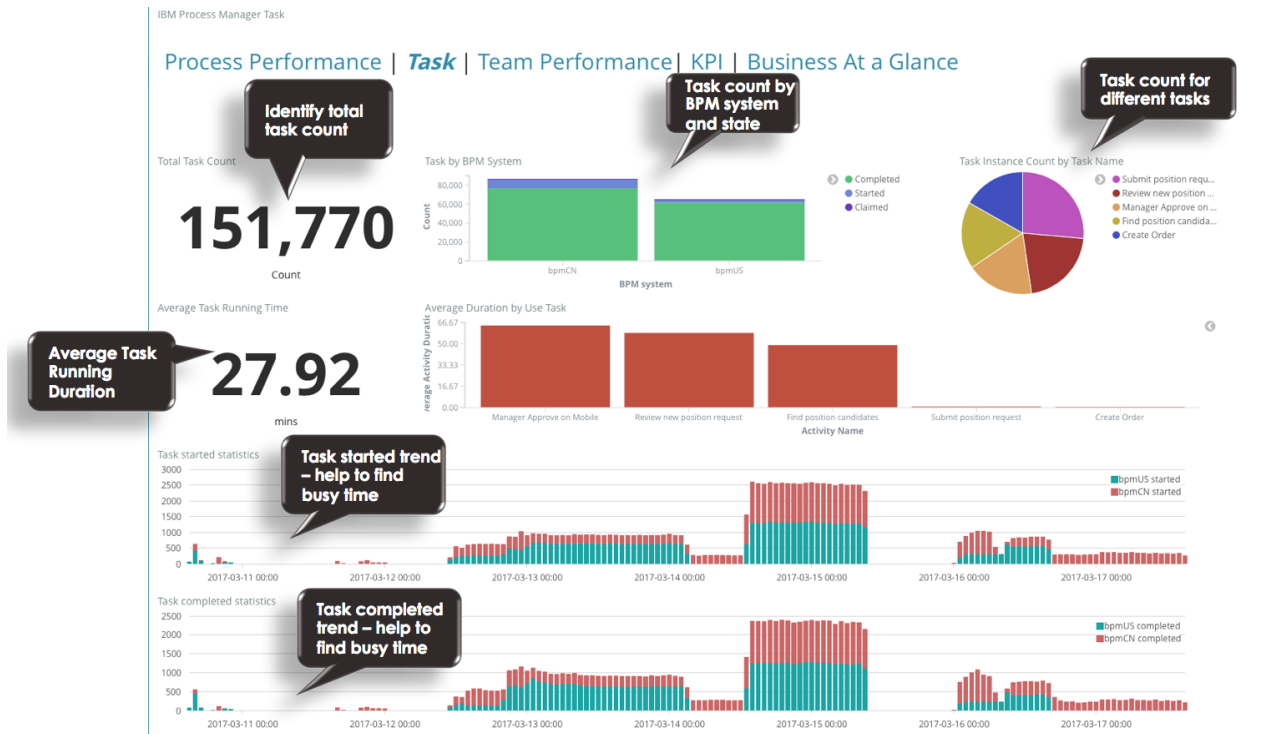
Name ▲
IBM BPM BusinessData
IBM BPM KPI
IBM BPM Process Performance
IBM BPM Task
IBM BPM TeamPerformance

1. Select **Process Performance** to switch to the Process Performance dashboard which is the default dashboard to display process related statistics



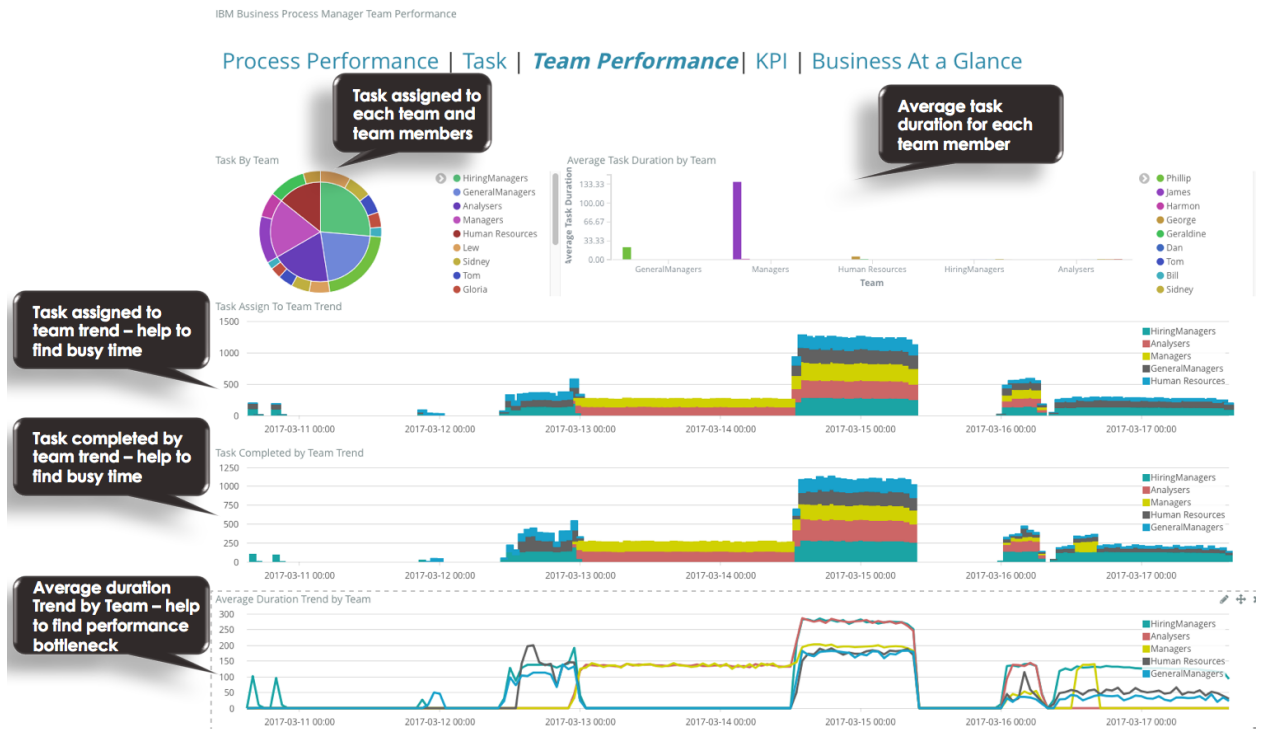
2. Select **Task** to switch to the Task dashboard which is the default dashboard to display user task related statistics.

**Note:** This dashboard is only for **User Task**. System Task and other activities are not taken into consideration.



3. Select **Team Performance** to switch to the Team Performance dashboard which is the default dashboard to display team related statistics.

**Note:** This dashboard is only for **User Task**. System Task and other activities are not taken into consideration.



4. Select **KPI** to switch to the KPI dashboard which is an empty dashboard. You can add your custom KPI visualization here.

IBM Business Process Manager KPI

Process Performance | Task | Team Performance | **KPI** | Business At a Glance

5. Select **Business At a Glance** to switch to the Business Data dashboard. You can add your custom business data visualization here.

IBM Business Process Manager BusinessData

Process Performance | Task | Team Performance | KPI | **Business At a Glance**

## Import the Hiring Sample Dashboard (Optional)

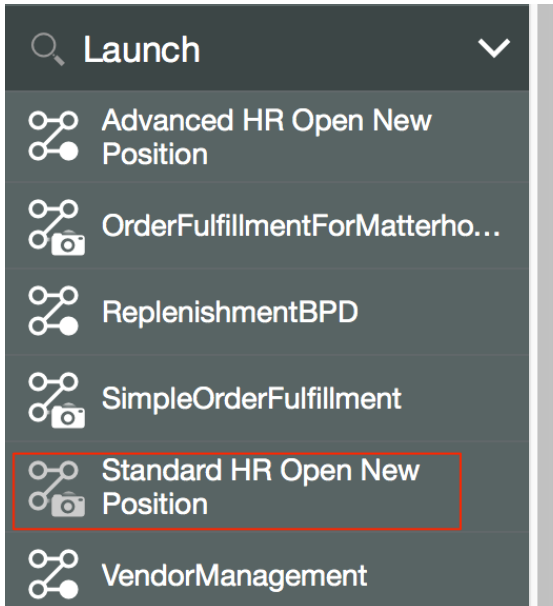
Included is a sample Business Data dashboard for the Hiring Sample. To import it, follow these steps:


1. Modify **EventSummaryAgent.yml** for EventSummaryAgent (EventSummaryAgent/conf/EventSummaryAgent.yml) to enable the sample Business Data processor

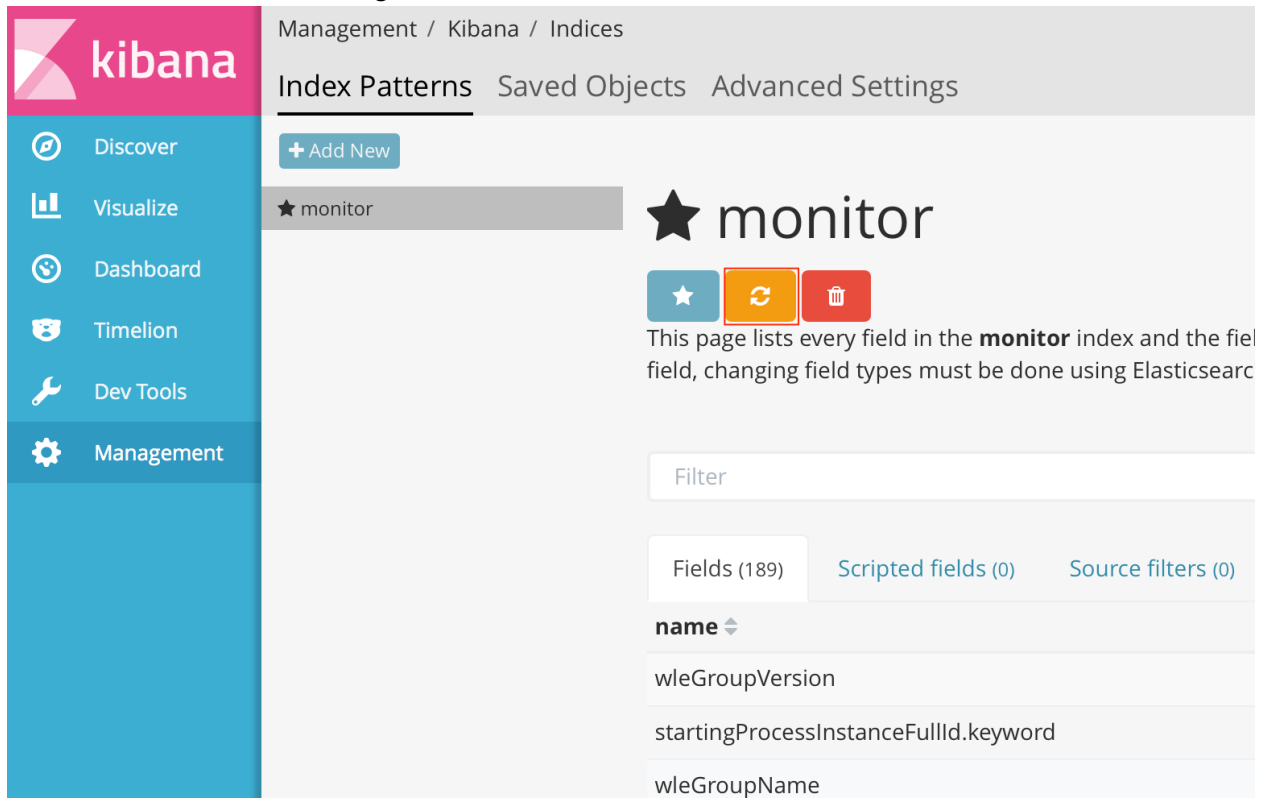
```
# Enable the sample Business Data aggregator  
enableBusinessDataAggregator:true
```

2. Restart EventSummaryAgent.
3. In BPM Process Portal, run a process instance for Standard HR Open New Position in Hiring Sample and **COMPLETE** it.

**Note:** You need to **complete** a process instances to ensure that all of the raw data of Elasticsearch index is included.

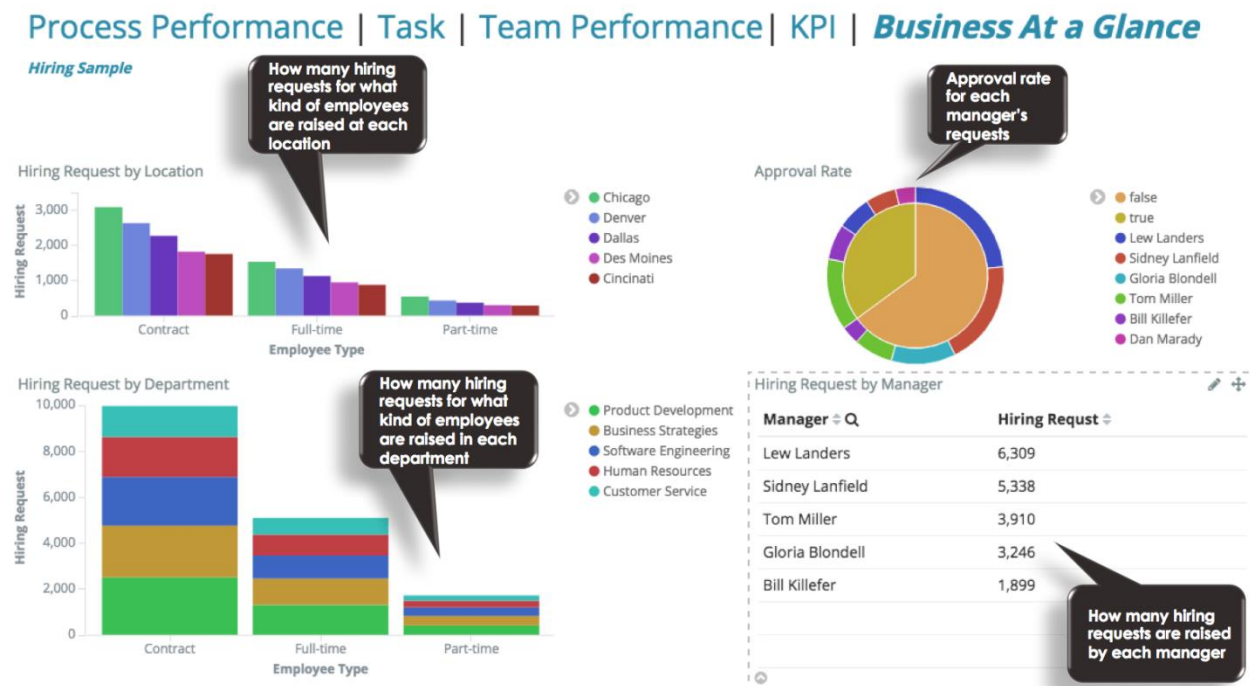


- Click **Management > Index Patterns**, select **monitor** index, click . Kibana will retrieve the new fields for Business Data that are generated in the Elasticsearch index.



The screenshot shows the Kibana Management interface. On the left is a sidebar with navigation links: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management (highlighted). The main header shows 'Management / Kibana / Indices' and tabs for 'Index Patterns', 'Saved Objects', and 'Advanced Settings'. Below the tabs is a '+ Add New' button and a list of index patterns, with 'monitor' selected and marked with a star. The right pane displays the 'monitor' index details, including a star icon, a refresh icon, and a delete icon. Below these icons is a text description: 'This page lists every field in the **monitor** index and the field, changing field types must be done using Elasticsearch'. A 'Filter' input field is present. Below the filter are three tabs: 'Fields (189)', 'Scripted fields (0)', and 'Source filters (0)'. The 'Fields (189)' tab is active, showing a list of fields: 'name', 'wleGroupVersion', 'startingProcessInstanceFullId.keyword', and 'wleGroupName'.

- In Kibana, click **Management > Saved Objects**, click **Import**, select **HiringSampleDashboard.json**.
- Click **Business At a Glance** to see the custom Business Data dashboard for the Hiring Sample.



## Handling the Tracked Business Data

Through IBM BPM Process Designer, the fields of the process variables can be declared as “Performance Tracking”. The tracked fields belonging to this process will be recorded as a Tracking Group automatically if the **Enable Autotracking** check box has been selected.

For example, if the following tracked fields have been enabled:

The screenshot shows the 'Tracking' tab in the IBM BPM Process Designer. Under 'Tracking Groups', 'Enable Autotracking' is checked, and the 'Autotracking Name' is 'AT\_linked\_loan\_record'. Under 'Auto-Tracked Fields', four fields are listed: 'loanAmount', 'loanee.name', 'loanee.age', and 'status'. Each field has a corresponding input box and a delete icon (X).

Field Name	Input Box	Delete Icon
loanAmount	loanAmount	X
loanee.name	loanee_name	X
loanee.age	loan_age	X
status	status	X

In this situation, the DEF event will include the tracked fields selected in the Process Designer and Activity Event will have the attribute name “trackedFields” which include the tracked fields. For example:

```
"trackedFields": {  
  "loanAmount.integer": 2000,  
  "loanee_name.string": "Brian",  
  "loan_age.integer": 35,  
  "status.string": "Origination"  
}
```

## Enabling the business data aggregator

By default, the business data aggregator is enabled at the EventSummaryAgent configuration file. Open the EventSummaryAgent/conf/EventSummaryAgent.yml to check that the enablement is set to “true”:

```
# Enable the sample Business Data aggregator  
enableBusinessDataAggregator:true
```

Enabling the business data aggregator will create a new type of “BusinessData” at the Elasticsearch index and the Process Summary event will include the business data also.

For one process instance, it has one document for all tracking groups. If the tracked field has been updated at the process flow, the document will reflect the latest value.

When the process includes a sub or linked process, the sub or linked process belongs to the main process and it still creates one single document for the business data. For example:

```
"trackedFields": {  
  "AT_Loan_record1495615322291": {  
    "loanAmount.integer": 2000,  
  }  
}
```

```

    "region.string": "UK",
    "status.string": "Origination",
    "loanee_name.string": "Brian",
    "loanee_age.integer": 35
  },
  "AT_linked_loan_record": {
    "loanAmount.integer": 2000,
    "loanee_name.string": "Brian",
    "loan_age.integer": 35,
    "status.string": "Origination"
  }
}

```

As the example shows, the AT\_Loan\_record1495615322291 is the tracking group name which is belong to the main process, and the AT\_linked\_loan\_record is the linked process' tracking group name.

### Correlating the tracking groups

When using the sub or linked process, the tracked fields at the main process and the sub/linked process will be stored in the different tracking groups, but sometimes logically, these tracking fields might stand for the same business entity.

In this situation, you may need to correlate the tracking groups which belong to the same starting process instance.

As the configuration example shows, you can configure the correlation field name per application and per BPD process.

#### NOTE:

1. Assigning multiple correlation fields for the same process and same application is not supported
2. Cross process instance correlation is not supported
3. Use the field name as the correlation, the value of the fields will be decided by the latest generated activity event
4. Wildcard (\*) is accepted as the processApplicationName or processName

#### Configuration example:

In the "Bank CC Loan (sub)" process in the "Test Emit Monitor Event" application, use the tracked field "loanee\_name.string" as the correlation. Meanwhile, in the "Test Order" application, all processes use the tracked field "order\_id.string" as the correlation.

**Note:** The tracked fields need to include the type "postfix" because defining the track field in different tracking groups using the same name, but having a different type, means that the tracked field cannot be correlated.

```

innerProcessCorrelation:
- processApplicationName: Test Emit Monitor Event
  processName: Bank CC Loan (sub)
  correlationFieldName: loanee_name.string
- processApplicationName: Test Order
  processName: '*'
  correlationFieldName: order_id.string

```



After setting the correlation field `loanee_name.string`, the example at the section - *Enable business data aggregator* will be merged to:

```
"mergedTrackedFields": {
  "loanAmount.integer": 2000,
  "status.string": "Origination",
  "loanee_name.string": "Brian",
  "loan_age.integer": 35,
  "loanee_age.integer": 35,
  "region.string": "UK"
}
```

The `loan_age.integer` and `loanee_age.integer` at the different tracking groups cannot be merged to single tracked field, because the name is different. For the other tracked fields at the different tracking groups and with the same name, it will be merged to single value.

## Drill in at the “Process Performance” and “Task” dashboard

When the `enableBusinessDataAggregator` has been set to `true` at the `EventSummaryAgent` configuration file, the **Process Performance** and **Task** and **Team Performance** dashboard support drill in by the pre-defined tracked fields as dimension. You can input the filter in the head of the dashboard, for example:

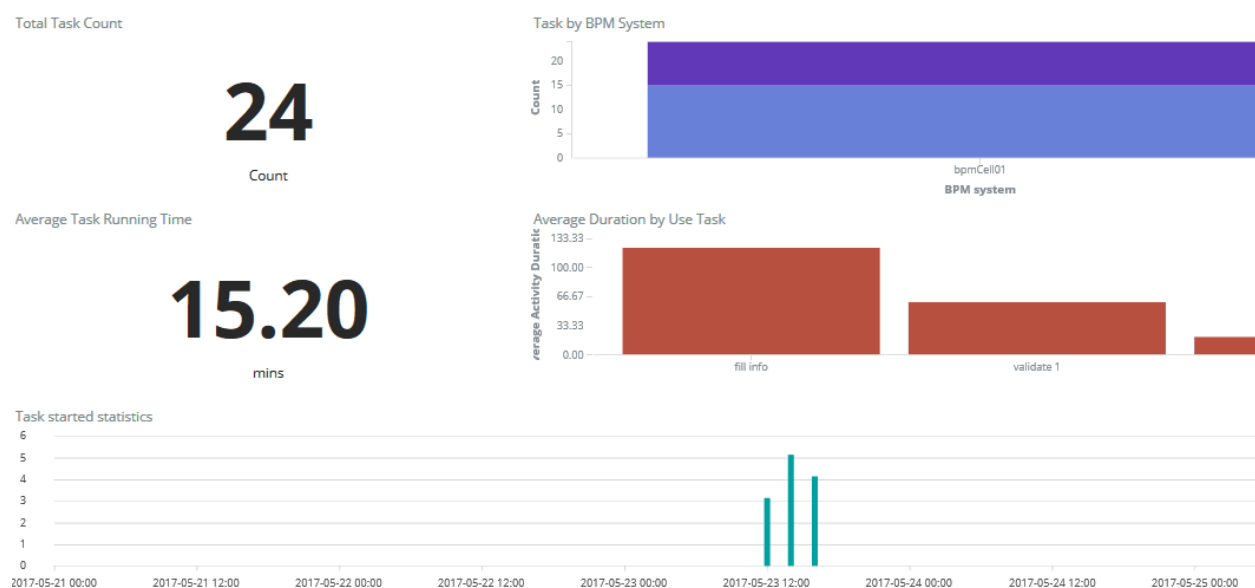
If you only want to show the Task status for one process application like “Test Emit Monitor Event”, you can input the **`processApplicationName: "Test Emit Monitor Event"`** as the filter, the dashboard will reflect the filter immediately:

IBM BPM Task

processApplicationName:"Test Emit Monitor Event"

IBM Process Manager Task

[Process Performance](#) | [Task](#) | [Team Performance](#) | [KPI](#) | [Business At a Glance](#)



**Note:** Because Kibana is powered by Elasticsearch it supports the powerful Lucene Query String syntax, as well as making use of some of Elasticsearch's filter capabilities. If you want to drill in to the process application "Test Emit Monitor Event" and only focus on the tasks which related with the pre-defined tracked field "Region" equals to "UK", you can input the filter as:

processApplicationName:"Test Emit Monitor Event" AND trackedFields.region.string:UK

## Troubleshooting

---

### Troubleshooting when the event was not generated

If you find the dashboard at Kibana was not updated correctly or has no new events coming in, you need start from the **BPMEventEmitter** to **EventSummaryAgent** section of this document to troubleshoot.

1. Check if the DEF event can be generated by IBM BPM
  - Stop the BPMEventEmitter temporarily. Use the WAS admin console, select **Service integration > Service Integration Bus Brower > Your\_BUS\_name> Destinations > Your\_DEF\_destination > Queue points**
  - At this page, because the DEF event consumer – BPMEventEmitter - has been stopped, you can see the DEF events stuck at this queue, and the queue depth is increasing.
  - Otherwise, check the steps in *Enable the Dynamic Event Framework (DEF)* to make sure DEF was successfully enabled.
2. Check if the ProcessEvent and ActivityEvent document can be generated at the Elasticsearch
  - Get the basic count of ProcessEvent by running the following command at Kibana's Dev Tools:  

```
GET monitor/ProcessEvent/_count
```

**Note:** the monitor is the default index name, which is configurable, replace it according to your configuration.
  - Get the basic count of the ActivityEvent by running the following command at the Kibana's Dev Tools:  

```
GET monitor/ActivityEvent/_count
```
  - Start the BPMEventEmitter, after this step, you should see the count of the ProcessEvent and ActivityEvent increase. Otherwise, check the BPMEventEmitter's log, which belongs to the SystemOut.log at the server host the BPMEventEmitter WAR.
3. Check if the ProcessSummary and ActivitySummary type document can be generated at the Elasticsearch
  - Stop the EventSummaryAgent
  - Get the basic count of ProcessSummary by running the following command at the Kibana's Dev Tools:

```
GET monitor/ProcessSummary/_count
```

- Get the basic count of the ActivitySummary by running the following command at the Kibana's Dev Tools:

```
GET monitor/ActivitySummary/_count
```

- After the ProcessEvent and ActivityEvent document count increases, you can start the EventSummaryAgent to see if it works.

After the EventSummaryAgent has been started (by default it is one minute) the EventSummaryAgent will start to pick up raw event type documents and the number of the combined type document will increase. Check the logs at the `<EventSummaryAgent_Root>/log` folder if you found the combined type documents has not increased.

## Troubleshooting the exceptions

Ensure that you change the WAR module's "Class loader order" to "Classes loaded with local class loader first (parent last)" if either of these exceptions are received in the SystemOut.log:

Exception 1	CNTR0020E: EJB threw an unexpected (non-declared) exception during invocation of method "onMessage" on bean "BeanId(BPMEventEmitter_war#BPMEventEmitter.war#BPMEventEmitterMDB, null)". Exception data: java.lang.NoSuchMethodError: org/apache/http/HttpHost.create(Ljava/lang/String;)Lorg/apache/http/HttpHost;
Exception 2	0000b44b LifecycleManager E class com.ibm.bpm.mon.oi.LifecycleManagerBeanstartMethod E: Create elastic search index mapping failed with exception. java.io.IOException: listener timeout after waiting for [10000] ms

## Potential event lost

If your DEF queue depth did not configure properly, it has the potential for the event to be lost.

The default queue depth: The high message threshold is 50000. If your BPMEventEmitter was not started or the event producer created the events faster than the consumer, the events which exceed the threshold will be lost.

Tune this number to match your system's peak time to avoid failure of the event consumer.

For more information of this exception, see:

<http://www-01.ibm.com/support/docview.wss?uid=swg21624736>

## Maintaining the indices

---

### Maintaining the scale of the index

As defined by the BPM Analytics system admin, you might not want to keep all of the historical data at the single ES index. One option is to create separated indices by the time ranges. For example, one index per month or per quarter, depends on the scale of events the BPM system created.

The multiple ES indices can be wrapped by an index alias, for example: you have the index “monitor201703” and index “monitor201704”, you can use the following API to create an index alias “source”.

```
POST /_aliases
{
  "actions" : [
    { "add" : { "index" : "monitor201703", "alias" : "source" } },
    { "add" : { "index" : "monitor201704", "alias" : "source" } }
  ]
}
```

The index alias with multiple indices do not accept the write operation, it is read only. This means that the destination of the BPMEventEmitter and the EventSummaryAgent do not support this kind of the alias. However, as the source of the EventSummaryAgent, if you have multiple indices contain the events, it is proper to use an index alias. Continuing the above example, you can use the index alias “source” as the ES source configuration:

```
# the configuration properties for the source elastic search
# for the source event, the EventSummaryAgent only read the data without update operation
# that can make sure it's safe to use ES index alias with multiple indices configured
esSourceConfiguration:
  hosts: localhost:9200
  index: source
```

### Rebuild the event summary index

As long as you have the raw events which created by the BPMEventEmitter, you can rebuild the summary event index at any time.

For example, you have the configuration file of the EventSummaryAgent:

```
# the configuration properties for the source elastic search
# for the source event, the EventSummaryAgent only read the data without update operation
# that can make sure it's safe to use ES index alias with multiple indices configured
esSourceConfiguration:
  hosts: localhost:9200
  index: source
# the following properties should be enabled when elastic search security is on
username: elastic
password: <xor>d3hrdXJEeXU=
httpsTrustType:
trustFileLocation:
hostnameVerifier:
# the configuration properties for the target elastic search
```

```

# write the combined summary event as the target
# Notes: the index alias with multiple indices do not support write operation
esTargetConfiguration:
  hosts: localhost:9200
  index: monitor
  # the following properties should be enabled when elastic search security is on
  username: elastic
  password: <xor>d3hrdXJEeXU=
  httpsTrustType:
  trustFileLocation:
  hostnameVerifier:

# the progress will be recorded at the target Elasticsearch
progressStorage:
  # Index name used to store progress data
  esIndex: oiprogress
  # Type name used to store progress data
  esType: readcursor

```

You can rebuild the event summary index by the following procedure:

1. Stop the EventSummaryAgent
2. Delete the event summary index “monitor”
3. Delete the progress record index “oiprogress”
4. Restart the EventSummaryAgent

## Archive and restore

To prevent data loss when Elasticsearch crashes or when data has been deleted by accident, BPM analytics supports archiving the raw events from the Elasticsearch to certain object store. And, when needed, the data can be restored from the object store to Elasticsearch by using EventSummaryAgent.

### Configurations for archive and restore

Configurations below in EventSummaryAgent configuration file is related to archive and restore function.

```

# archive related configurations
archive:
  enabled: false
# In object store we will merge several raw events into one object file
# minBulkSize controls the min events count for a bulk when possible
# maxBulkSize controls the max events count for a bulk.
# You should adjust that according to the event average size. It cannot bigger than
10000
  minBulkSize: 100
  maxBulkSize: 1000
# the configuration properties for the object storage
# we support object storage which use openstack-swift APIs
# Because different authentication method may used by openstack-swift
# So different setting sets may used
objectStoreConfiguration:

```

```

# Set 1. Sample for using openstack swift interface with tempauth
#=====
providerOrApi: swift-tempAuth
identity: test:tester
credential: testing
endpoint: http://localhost:8080/auth/v1.0
#You can assign the container name where store the files.
#If no the default one will be used
containerName: BPMAnalyticArchive
#swift need location to be settled. If no the first location we find will be used
#location: reginOne
#If you are using private object storage without valid certification for HTTPs.
You should
#enable below settings:
#overrides:
#   jclouds.relax-hostname: true
#   jclouds.trust-all-certs: true
#=====
#Set 2. Sample for using openstack swift v3 auth (IBM Bluemix object storage
supported)
#=====
#providerOrApi: swift-v3
#containerName: BPMAnalyticArchive
#userId: 3eef42de3e3*****
#password: L21i*****
#auth_url: https://identity.open.softlayer.com
#domainName: 136****
#project: object_storage_1c8707e5_2480_****_b6ca_*****
#If you are using private cloud without valid SSL certification for HTTPs. Set
below parameter to true.
#disableSSLVerification: true
#=====

```

**enabled:** Controls if the archive function enabled or not.

**minBulkSize/maxBulkSize:** Controls the event count saved in one object store file (object).

**objectStoreConfiguration:** Object store configurations. There are too many kinds of object stores used on the internet as public or private cloud services. We only now support two of them - openstack-swift API with v1.0 temp auth and v3.0 keystone auth. Refer to configuration file comments for configuration details. The configuration in the sample is for swift temp auth method. The sample below are for Bluemix object store:

```

providerOrApi: swift-v3
containerName: BPMAnalyticArchive
userId: 3eef42de3xxxxxxxxxxxxx
password: xxxxxxxx
auth_url: https://identity.open.softlayer.com
domainName: 136xxx
projectId: object_storage_1c8707e5_xxxx_xxxx_xxxx_9c9xxxxxxxx1

```

## How to enable archiving and when it happens

When you set enabled to “true” under archive and have configured the objectStoreCofniguration, you can start the EventSummaryAgent with that configuration. The EventSummaryAgent will save the data to the object store when there is no jobs for the aggregation:

```
Jun 15, 2017 10:02:48 AM com.ibm.bpm.mon.oi.combine.EventSummaryAgent doJob
INFO: I: Start getting process/activity events from the data source.
Jun 15, 2017 10:02:49 AM com.ibm.bpm.mon.oi.archive.ArchiveHandler run
INFO: I: Archive process started.
Jun 15, 2017 10:02:49 AM com.ibm.bpm.mon.oi.archive.ArchiveHandler run
INFO: I: Archive process finished.
Jun 15, 2017 10:02:49 AM com.ibm.bpm.mon.oi.combine.EventSummaryAgent doJob
INFO: I: Start getting process/activity events from the data source.
Jun 15, 2017 10:02:49 AM com.ibm.bpm.mon.oi.archive.ArchiveHandler run
```

## How to restore events to Elasticsearch

The restore process can be started by run EventSummaryAgent with -restore parameter.

**Note:** You must pay attention that the restore target is controlled by esSourceConfiguration at the EventSummaryAgent’s configuration and esConfiguration.index at the BPMEventEmitter’s configuration, because the EventSummaryAgent’s Elasticsearch source connection configuration might use the index alias which is read-only. For the details about the index alias, see the section – *Maintaining the indices* in this document. Since the esSourceConfiguration.index might be read-only, so restore operation use the connection information from the esSourceConfiguration in EventSummaryAgent, and use the index name settled in BPMEventEmitter.

When BPMEventEmitter starts it will save the configuration into Elasticsearch index configured by configuration:

**esTaskIndex:** restore\_task\_index

The restore process will read configuration in that index. You can double check before restoring the archived events.

The restore is controlled by the time range. You can put task json in esTaskIndex with type ‘restore\_task\_type’ in the same format below:

```
{"startTime":"YYYY-MM", "endTime":"YYYY-MM"}
```

For example:

```
POST restore_task_index/restore_task_type
{
  "startTime":"2017-06",
  "endTime":"2017-06"
}
```

In the sample task, all events in 2017-06 will be restored from the object store Elasticsearch if any.

We support month only, and you can assign months across years as below:

```
POST restore_task_index/restore_task_type
{
  "startTime":"2016-01",
  "endTime":"2017-01"
}
```

That will restore all the events in year 2016 and events in 2017-01.

You can start the restore process as below:

```
EventSummaryAgent.bat/sh -restore
IBM BPM Event Summary Agent

Jun 16, 2017 3:40:36 PM com.ibm.bpm.mon.oi.restore.RestoreWorker restore
INFO: I: Restore process is running.
Jun 16, 2017 3:40:41 PM com.ibm.bpm.mon.oi.restore.RestoreWorker restore
INFO: I: Restore process finished.
Jun 16, 2017 3:40:41 PM com.ibm.bpm.mon.oi.combine.EventSummaryAgent doJob
INFO: I: Start getting process/activity events from the data source.
```

You can start restoring with specific configuration file locations also:

```
EventSummaryAgent.bat/sh -configFile <configurationFileLocation> -restore
```

After the restore tasks at the esTaskIndex finishes, the EventSummaryAgent will return to normal aggregation process automatically. This helps to avoid System Admin interaction for a lengthy restore job.

**Note:** Before you start the EventSummaryAgent at the restore mode, ensure that you have put all of the restore tasks at the Elasticsearch index, or the EventSummaryAgent will switch to the aggregation mode immediately because there is no restore task.