IBM Prescriptive Quality for Manufacturing Private

# User Guide

*Version 1 Release 2*

IBM Prescriptive Quality for Manufacturing Private

# User Guide

*Version 1 Release 2*

# Note

Before using this information and the product it supports, read the information in "Notices" on page 77.

First edition (August 2018)

# Contents

# Chapter 1. Product overview

IBM® Prescriptive Quality for Manufacturing Private helps detect and prioritize quality problems earlier and more definitively than can be done using traditional statistical process control techniques. Designed for lines of business that have quality responsibilities across supply chain and manufacturing, it can help improve production yield, increase throughput, and reduce overall costs associated with poor quality.

## Welcome to Prescriptive Quality for Manufacturing Private

IBM Prescriptive Quality for Manufacturing Private includes a parametric entity and an inspection entity. The parametric entity detects unfavorable changes in variable-type data, and provides information that facilitates diagnostics and alarm prioritization. The inspection entity detects unfavorable changes in the quality of components.

IBM Research's Quality Early Warning System (QEWS) algorithm in Prescriptive Quality for Manufacturing Private detects emerging quality problems sooner and with fewer false alarms than is typically achieved by traditional statistical process control. To achieve earlier detection, QEWS is sensitive to subtle changes in data values, such as shifts that are small in magnitude or trends that grow slowly over time. For a given level of statistical confidence, QEWS typically needs fewer data points than traditional statistical process control.

Early detection of quality problems is essential where delayed detection can have significant negative consequences, such as in the following scenarios:

- Building a large inventory of defective products results in high scrap costs.
- Having widespread quality or reliability problems in the field results in damage to brand value.
- Compromised production of supply-constrained materials or components prevents on-time shipment.
- Compromised production of products with long manufacturing times results in shipment delays.

### Parametric entity

In the parametric entity, Quality Early Warning System for Variable Data (QEWSV) monitors variable-type data. Variables are defined for every operation per tool. Variables are equated with Measurement type, whose measurements are read at different time intervals during the sequence of the operation flow. This type of data is found in several industrial applications, including Supply Chain, Manufacturing, and Finance applications.

QEWSV identifies unfavorable trends in the data process. The focus is on providing timely detection of unacceptable process behavior while maintaining a pre-specified low rate of false alarms.

Variable values and evidence charts are plotted by using parametric results. The deviation or drift from target values are computed and analyzed to show whether the process sequence is adhering to normal operation limits.

## Inspection entity

In a manufacturing environment, defects can occur in a manufacturing process because of variations in factors like process, raw materials, design, and technology. The resulting low quality of products creates a larger inventory of defective lots, which leads to increased inspection effort.

A small delay in detecting a quality problem can result in large costs, lost opportunity, and lost brand value.

In the inspection entity, QEWS evaluates evidence to determine whether the rate of failures is at an acceptable level. QEWS highlights combinations for which the evidence exceeds a specified threshold. QEWS can detect emerging trends earlier than traditional statistical process control, such as trend analysis. QEWS maintains a specified low rate of false alarms. Post-warning analysis of charts and tables identifies the point of origin, the nature and severity of the problem, and the current state of the process.

The inspection entity analyzes data from the inspection, testing, or measurement of a product or process operation over time. The data can be obtained from the following sources:

- Suppliers (for example, the final manufacturing test yield of a procured assembly)
- Manufacturing operations (for example, the acceptance rate for a dimensional check of a machined component)
- Customers (for example, survey satisfaction ratings)

*Products* are the subjects of QEWS analyses. A product is typically a part or a part assembly, but it can also be a process or a material. Products might be used in larger finished assemblies, which QEWS calls resources. A product can be associated with any resource, process, material, location, or a combination of these entities during inspection analysis.

You can adjust the frequency at which data is captured and input to QEWS, and the frequency at which QEWS analyses are run, according to the requirements of each situation. For example, monitoring the quality levels of assemblies that are procured from a supplier might best be done on a weekly basis; monitoring the quality levels of units that are moving through a manufacturing operation might best be done on daily basis.

# What's new

The What's new section contains a list of new, changed, and deprecated configuration features for this release. It also contains a cumulative list of similar information for previous releases. It will help you plan your application deployment strategies and the training requirements for your users.

## New features

### New features in the August 2018 release

The following new features were added in this release:

- New API call: Trigger target setting for QEWS attribute analysis. Triggers the target setting procedure for IBM Quality Early Warning System analysis for attributes. The analysis uses event data and parameter data from IBM Prescriptive Quality for Manufacturing Private.

- Authentication using the provisioning console.
- Support for data formats of previous and current versions of the Watson™ IoT Platform.

**New features in the November 2017 release**

Four new API calls were added in this release:

**Enable QEWS multivariate analysis on multiple variables independently**
Enables the IBM Quality Early Warning System multivariate analysis for the data for multiple variables independently from your other data on IBM Prescriptive Quality for Manufacturing.

**Enable target setting for QEWS variable analysis on a variable independently**
Enables you to perform the target setting procedures for IBM Quality Early Warning System variable analysis for a variable independently from your other data on IBM Prescriptive Quality for Manufacturing.

**Trigger target setting for QEWS variable analysis**

Triggers the target setting procedure for IBM Quality Early Warning System analysis for variables. The analysis uses event data and parameter data from IBM Prescriptive Quality for Manufacturing.

**Get target setting for QEWS variable analysis**
Gets a set of variable target setting results from previous target setting runs.

**New features in the September 2017 release**

- REST API. The application programming interface provides actions such as enabling or triggering an IBM Quality Early Warning System analysis on a variable or attribute, setting or removing an analysis interval, and getting the analysis results.

## Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products.

For information about the commitment that IBM has to accessibility, see the IBM Accessibility Center (www.ibm.com/able).

HTML documentation has accessibility features. PDF documents are supplemental and, as such, include no added accessibility features.

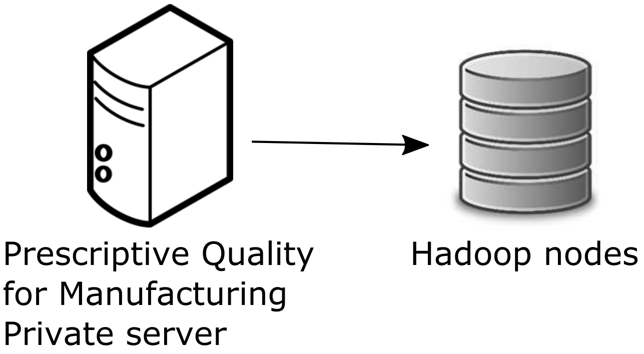
# Chapter 2. Installing the product

Before using Prescriptive Quality for Manufacturing Private, you must install the server and artifacts. Follow these instructions to install the product or to upgrade an existing installation to the latest version.

## Prescriptive Quality for Manufacturing Private deployment topology

The topology of Prescriptive Quality for Manufacturing Private consists of the Prescriptive Quality for Manufacturing Private server and Hadoop nodes.

The following image shows the deployment topology of Prescriptive Quality for Manufacturing Private.



### Prescriptive Quality for Manufacturing Private server

The Prescriptive Quality for Manufacturing Private server is an application server. The cloud server profile and provisioning console profile run on it.

### Hadoop nodes

The Prescriptive Quality for Manufacturing Private server accesses the Hortonworks Data Platform nodes to read and write data.

Hortonworks Data Platform 2.6.1 and Ambari 2.5.1 are supported by Prescriptive Quality for Manufacturing Private. HDFS, HBase, and Spark are the components for using Prescriptive Quality for Manufacturing Private.

For more information about the hardware requirements for Apache Hadoop, refer to the Hortonworks Data Platform hardware recommendation site.

For more information about how to install the Hortonworks Data Platform 2.6.1 cluster, refer to the Hortonworks Data Platform documentation site.

## System requirements

Before installing Prescriptive Quality for Manufacturing Private, ensure that the target system meets the system requirements.

The computer where you run the installation must have sufficient disk space that is allotted to the / directory. The minimum disk space for the / directory is 100GB.

The other system requirements for the installation server are as follows:

- 4-core processor
- 32GB memory
- Red Hat Enterprise Linux Server Edition version 7.2, x86_64.

The system that you use for the Prescriptive Quality for Manufacturing Private server must meet the following requirements:

- 4-core processor
- 32GB memory
- 500GB hard disk
- Red Hat Enterprise Linux Server Edition version 7.2, x86_64.

## Upgrading the product

If you upgrade from the previous version of Prescriptive Quality for Manufacturing Private, you must only follow the steps in the Installing the artifacts task.

## Prerequisites and installation preparation

Complete these tasks before you install Prescriptive Quality for Manufacturing Private.

### Configuring passwordless authentication

After you create an instance of Prescriptive Quality for Manufacturing Private, you must configure passwordless SSH authentication from Prescriptive Quality for Manufacturing Private to all Hortonworks Data Platform nodes in your environment.

#### Procedure

1. Log in as `root` to the server that hosts Prescriptive Quality for Manufacturing Private.
2. Generate an SSH authentication key pair:
   ```
   ssh-keygen -t rsa -b 2048
   ```
3. Copy the public key to an Hortonworks Data Platform node and edit the `/etc/sudoers` file for that node. The following steps use the `ambari` node as an example:
   a. Run the following command to copy the public key to the node:
      ```
      ssh-copy-id -i ~/.ssh/id_rsa.pub root@ambari
      ```
   b. Run the following command to log in without using strict host key checking:
      ```
      ssh -o "StrictHostKeyChecking no" root@ambari
      ```
   c. Launch vi with superuser privileges. If you cannot use `visudo`, use `sudo`.
      ```
      visudo su
      ```
   d. Change the permissions of the `/etc/sudoers` file:
      ```
      chmod 640 /etc/sudoers
      ```
   e. Edit the `/etc/sudoers` file:
      ```
      vi /etc/sudoers
      ```

f. Find the following line in the file:

```
Defaults requiretty
```

Change this line to:

```
#Defaults requiretty
```

g. Quit vi.

h. Change the permissions of the /etc/sudoers file:

```
chmod 440 /etc/sudoers
```

Repeat these steps to copy the public key to all Hortonworks Data Platform nodes.

## Preparing for installation

Complete the following steps before you install Prescriptive Quality for Manufacturing Private.

### Procedure

1. On the system on which you will install Prescriptive Quality for Manufacturing Private, open the /etc/hosts file.
2. Ensure that your Hortonworks Data Platform nodes are listed in the file, for example:

```
127.0.0.1 localhost.localdomain localhost
##.##.##.## hdpmgmt01.domain.com hdpmgmt01
##.##.##.## hdpmgmt02.domain.com hdpmgmt02
##.##.##.## hdpslave01.domain.com hdpslave01
##.##.##.## hdpslave02.domain.com hdpslave02
##.##.##.## hdpslave03.domain.com hdpslave03
```

3. Save and close the file.
4. Access the Ambari web user interface from a web browser by using the server name (the fully qualified domain name) on which you installed the Ambari server, and port 8080. For example, enter the following string in your browser: HTTP://*node1.example.com*:8080
5. Select **Add New Hosts** and go to the Install Options page. In **Target Hosts**, list the node where you will install Prescriptive Quality for Manufacturing Private.
6. In **Host Registration Information**, select one of the following options:
   - Provide your SSH Private Key to automatically register the host:

     Click **SSH Private Key**. The private key file is /root/.ssh/id_rsa, if the root user installed the Ambari server. If you installed as a non-root user, then the default private key is in the .ssh directory in the non-root user's home directory.

     Click **Register and Confirm**.
   - Register the host manually and do not use SSH.
7. On each Hortonworks Data Platform management node and slave node, create OS-level users and groups by using the following commands:

```
groupadd pq
groupadd qews
useradd -g pq -s /bin/bash pq
echo "pq:pw@pq" | chpasswd
useradd -g qews -s /bin/bash qews
echo "qews:pw@qews" | chpasswd
```

8. If Kerberos is enabled in the Hortonworks Data Platform cluster, you must create a principal for Prescriptive Quality for Manufacturing Private in the Kerberos Key Distribution Center (KDC), generate a related keytab file, and

distribute the keytab file to all of Hortonworks Data Platform cluster nodes and the Prescriptive Quality for Manufacturing Private node.

a. On the node where the KDC is installed, run the following commands:
```
kadmin.local
kadmin.local: addprinc -pw pw@pq pq
kadmin.local: ktadd -k /etc/security/keytabs/pq.keytab -norandkey pq
kadmin.local: addprinc -pw pw@qews qews
kadmin.local: ktadd -k /etc/security/keytabs/qews.keytab -norandkey
qews
kadmin.local: exit
```
b. Copy the `/etc/security/keytabs/pq.keytab` file and the `/etc/security/keytabs/qews.keytab` file to the `/etc/security/keytabs` directory on all of Hortonworks Data Platform cluster nodes and the Prescriptive Quality for Manufacturing Private node.

## Installing the server

Perform this task to install the server component.

### Procedure

1. Download the `ServerInstallation.tar.gz`, `iotmf_server_1.2_l86-64_en.tar.gz`, and `ArtifactsInstallation.tar.gz` files to the system that you will use for the server. You can get `ServerInstallation.tar.gz` from the `IBM IOT for Manufacturing Foundation Server Installer 1.2 Linux x86-64 English` package. You can get `iotmf_server_1.2_l86-64_en.tar.gz` from the `IBM IOT for Manufacturing Foundation Server 1.2 Linux x86-64 English` package. You can get `ArtifactsInstallation.tar.gz` from the `Prescriptive Quality for Manufacturing Artifact Installer 1.2 Linux x86-64 English` package.
2. Copy the `ServerInstallation.tar.gz` file to the installation directory (*PQ_install_dir*) and decompress the file.

   The `ServerInstallation` directory is created.
3. Copy the `ArtifactsInstallation.tar.gz` file to the *PQ_install_dir* directory and decompress the file.
4. Copy the `iotmf_server_1.2_l86-64_en.tar.gz` file to the *PQ_install_dir*/`ServerInstallation/SolutionInstaller/NodeRoot/Downloads/Software` directory.
5. Run the following script to decompress the files:
   ```
   sh IOTMF_1.2_UncompressTarball.sh
   ```
6. To set up the installer, go to the installer setup folder *PQ_install_dir*/`ServerInstallation/SolutionInstaller` and then run the following script:
   ```
   ./setup.sh
   ```
7. Access the installer by going to the following URL:
   `https://`*`hostname`*`:8080/UI/index.html`
   where *hostname* is the host name for installer. Review the license information and then click **Accept**.
8. Click **Predefined Configuration** to load a predefined configuration that includes one node, WebSphere® Application Server Liberty, Provisioning Console, IBM JDK Artifact, and Service Framework.
9. Change the values in the **Property Editor** to the configuration of your server.
10. Click **Validate** to validate the configuration.
11. Click **Run** to run the installation.

12. In the installer setup folder, run the following script to clean up the installation files:
    ```
    ./cleanup.sh
    ```
13. Restart the server.

## Installing the artifacts

Perform this task to install the product artifacts.

### Procedure

1. To set up the installer, go to the installer setup folder and then run the following script:
   ```
   ./setup.sh
   ```
2. Access the installer by going to the following URL:
   `https://`*`hostname`*`:8080/UI/index.html`
   where *hostname* is the host name for installer. Review the license information and then click **Accept**.
3. Click **Predefined Configuration** to load a predefined configuration that includes one node and the **PQ Artifact** component.
4. Change the values in the **Property Editor** to the configuration of your server.
   - **Ambari host name** refers to the name of host where the Ambari server is installed. The artifact installer connects to the Ambari server to get Hortonworks Data Platform service status and connection information. If Kerberos is enabled in Hortonworks Data Platform, set **Kerberos is set as security authentication** to **True**.
   - The hbase keytab file and hdfs keytab file are files in the `/etc/security/keytabs` folder.
5. Click **Validate** to validate the configuration.
6. Click **Run** to run the installation.
7. In the installer setup folder, run the following script to clean up the installation files:
   ```
   ./cleanup.sh
   ```

## Post-installation tasks

Complete these tasks after you install Prescriptive Quality for Manufacturing Private.

### Creating system-level users

Before you first login to the product, you must create operating system-level users and groups on each Hortonworks Data Platform management node and slave node.

#### Procedure

1. Run the following command on each Hortonworks Data Platform management node and slave node:
   ```
   useradd -g pq -s /bin/bash -p pw@DEMOTENANT DEMOTENANT
   echo 'usermod -a -G qews DEMOTENANT'
   ```
2. If Kerberos is enabled in the Hortonworks Data Platform cluster, distribute the `/etc/security/keytabs/DEMOTENANT.keytab` file from the Ambari node to all other Hortonworks Data Platform nodes.

# Product license files

After you manually install IBM Prescriptive Quality for Manufacturing Private, you must ensure that the appropriate `swidtag` file and `license` directory exists on each computer on which you installed a Prescriptive Quality for Manufacturing Private component.

## License files

The `swidtag` file and `license` directory must exist on the Prescriptive Quality for Manufacturing Private node computer.

**Note:** Do not rename the directory or files.

If you installed on a Red Hat Enterprise Linux operating system by using the solution installer, the license files are automatically copied to /opt/IBM/PQ_1.2/license and the swidtag files are automatically copied to /opt/IBM/PQ_1.2/iso-swid on the Prescriptive Quality for Manufacturing Private node.

## Application licensing and the slmtag file

A licensing application runs on the Prescriptive Quality for Manufacturing Private Integration Bus node. The application periodically logs the number of assets that are deployed on the Prescriptive Quality for Manufacturing Private system into a file with the `.slmtag` extension in the `/opt/IBM/PQ_1.2` folder.

Here is a sample `slmtag` file with the following code.

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>3444c0728315463393b68a795ac7facd</PersistentId>
        <Name>IBM Prescriptive Quality for Manufacturing Private</Name>
        <InstanceId>/IBM/IOTM/PQ</InstanceId>
</SoftwareIdentity>
<Metric logTime="2017-08-24T09:07:28-05:00">
        <Type>VARIABLE</Type>
        <SubType>NO_VARIABLES</SubType>
        <Value>496</Value>
        <Period>
                <StartTime>2017-08-23T09:07:25-05:00</StartTime>
                <EndTime>2017-08-24T09:07:25-05:00</EndTime>
        </Period>
</Metric>
```

# Troubleshooting

Troubleshooting topics can help you identify and recover from common issues.

## Server validation fails with check_requiretty.sh error

Validation fails during the server installation, and the log file shows an error about check_requiretty.sh.

### About this task

This issue occurs because requiretty is enabled on the target machine. Complete the following steps to comment out the requiretty line in the `/etc/sudoers` file on the target system.

### Procedure

1. Run the following command:
   ```
   visudo
   ```
2. Comment out the requiretty line as follows:
   ```
   #Defaults requiretty
   ```
3. Save the file and exit.
4. Retry the server installation and validation.

## setup.sh fails with US-ASCII error

On a MacBook, setup.sh fails and the log file contains the following message: ERROR: ArgumentError: invalid byte sequence in US-ASCII.

### About this task

This error occurs because the default encoding of a MacBook is US-ASCII rather than UTF-8. Complete the following steps to change the encoding of a MacBook to UTF-8.

### Procedure

1. Run the following command on the MacBook:
   ```
   vi ~/.bash_profile
   ```
2. Add the following lines to the file:
   ```
   export LC_ALL=en_US.UTF-8
   export LANG=en_US.UTF-8
   ```
3. Save the file.
4. Run the following command:
   ```
   source ~/.bash_profile
   ```
5. Run the following command:
   ```
   echo $LANG
   ```
   If the result is `en_US.UTF-8`, the issue should be resolved.

## Cannot open the user interface for the installer in Firefox

The user interface for the installer does not open in Firefox when you use the following link: `https://`*`hostname`*`:8080/UI/index.html`

### About this task

Complete the following steps to remove the `cert8.db` file from Firefox and to try to open the user interface for the installer again.

### Procedure

1. In Firefox, select **Open menu**.
2. Select **Help** > **Troubleshooting Information**.
3. Select **Open Folder**.
4. Delete the `cert8.db` file.
5. Use the following link to reopen the user interface for the installer:
   ```
   https://hostname:8080/UI/index.html
   ```

# Chapter 3. Provisioning the product

Before using Prescriptive Quality for Manufacturing Private, you must provision the product. The provisioning process includes creating an instance of the product, creating an API key, creating one or more tenants, and adding users to tenants. You complete these tasks by using the provisioning console.

## Creating an instance

Before using Prescriptive Quality for Manufacturing Private, you must create an instance of the product. Only one instance of each product can be created.

### Procedure

1. Log in to the provisioning console user interface by using the following URL:
   `http://`*`hostname`*`:`*`port`*`/`
   where *hostname* is the provisioning console host and *port* is the port that is configured for the provisioning console.
2. Click **Instances**.
3. If no Prescriptive Quality for Manufacturing Private instance exists, create one:
   a. Click **Add Instance**.
   b. Select the IBM Prescriptive Quality for Manufacturing Private application type.
   c. Specify the application URL as:
   `https://`*`hostname`*`:`*`port`*`/ibm/pq/`where *hostname* is the Prescriptive Quality for Manufacturing Private host and *port* is the port that is configured for Prescriptive Quality for Manufacturing Private.
   d. Add the instance.

## Setting an API key

After you create an instance of Prescriptive Quality for Manufacturing Private, you must set an API key to enable the product to communicate with the provisioning console.

### Procedure

1. Log in to the provisioning console user interface by using the following URL: `http://`*`hostname`*`:`*`port`*`/`.
2. Click **Instances**.
3. Expand the Prescriptive Quality for Manufacturing Private instance to view the instance details.
4. Click **API Keys**. If no API key exists, click **Add**, then click **OK**.

## Creating tenants

After you create an instance of Prescriptive Quality for Manufacturing Private, you must create at least one tenant. A tenant is a group of users that are granted access to instances of one or more products. There is no limit to the number of users that can be assigned to a tenant. You can create one tenant for all your users, or you can create multiple tenants to group users and control their access to product instances. System administrators must log in as the tenant to add users for that tenant.

### Procedure

1. Log in to the provisioning console user interface by using the following URL: `http://hostname:port/`.
2. Click **Tenants**.
3. Click **Add Tenant**.
4. Enter an ID for the tenant. The ID must begin with a letter and contain only letters and numbers.
5. Enter an email address for the tenant. The tenant email address is used only as an identifier for the tenant. No email is sent to this address.
6. Enter a password for the tenant.
7. For tenant type, specify `internal`.
8. Leave the expiration date blank.
9. Leave the **Note** field blank.
10. Select the product instances that the tenant may access.
11. Click **Add**.

## Creating users

After you create an instance of Prescriptive Quality for Manufacturing Private and create at least one tenant, you can add users to the tenant. System administrators must log in as the tenant to add users for that tenant.

### Procedure

1. Log in to the provisioning console user interface by using the following URL: `http://hostname:port/`.
2. Click **Users**.
3. Click **Add User**.
4. Enter the username.
5. Enter the user's email address.
6. Enter a password for the user.
7. Enter the user's full name.
8. In the **Metadata** field, enter metadata in JSON format that indicates the application the user is allowed to access. For Prescriptive Quality for Manufacturing Private, enter the following metadata: `{'pq':''}`
9. Specify the tenant to which you want to add the user.
10. Click **Add**.

# Chapter 4. Using the product

You can use Prescriptive Quality for Manufacturing Private to upload data. With the data, you can view and analyze variables. You can also organize and track your variables.

## Logging on

Log on to access Prescriptive Quality for Manufacturing Private.

### About this task

Provide your IBM ID to your local administrator. Your administrator is responsible for ensuring that you have the security access level that is appropriate to your role in your organization. Your administrator supplies you with the web address URL for accessing the solution portal.

### Procedure

1. Enter the URL in the address field of the browser.

   **Note:** The fully qualified domain name is required in the URL, for example, `https://web_hostname:port/ibm/pq` where web_hostname is the host name of the web server. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly.
2. On the log in page, use the default ID and password. The default ID is *demouser*.

## Changing the login password

A system administrator can change the log in password for IBM Prescriptive Quality for Manufacturing Private. You can change the password if you want the password to be different from the default password.

### Procedure

To change the password, edit the server.xml file in the Prescriptive Quality for Manufacturing Private server installation folder.

## Downloading the tutorial and sample data

Prescriptive Quality for Manufacturing Private includes a tutorial and sample data.

### About this task

The topic shows how to download the tutorial and sample data for a process engineer.

### Procedure

1. Click the **User** icon and click **Tutorial download** to download the mp4 file for the steps for a process engineer.
2. Click **Sample data download** to download the sample data used for the tutorial.

## Uploading data files and previewing the data

On the home page, you can upload files for variable analysis and attribute analysis. Variable analysis and attribute analysis each require an event data file and an IBM Quality Early Warning System parameter file.

### About this task

When a file is uploaded, the system identifies the file type. The file types include; Variable Event file, IoT Variable Event file, Variable Parameter file, Attribute Event file, IoT Attribute Event file, and Attribute Parameter file. The IoT Variable Event file and the IoT Attribute Event file contain data that is collected from the Watson IoT Platform. The data is generated by the system automatically. When you upload a file, it is parsed to a file type according to the contents of the file.

Variable event files and attribute event files are used to identify variables and attributes. When they are identified, variable types and attribute types appear in the Object list. (Parameter files cannot be used to identify variables and attributes. Variables and attributes in parameter files must also be in event files to be correctly identified.)

The variable event file can contain dimensional data and must also contain variable values and one timestamp column. The timestamp column must use one of the following time formats:

- dd/MM/yyyy HH:mm:ss
- yyyy/MM/dd H:mm:ss
- yyyy/M/d H:mm:ss
- d/M/yyyy H:mm:ss
- dd.MM.yyyy HH:mm:ss
- MM-dd-yyyy HH:mm:ss tt, where tt = AM or PM
- M/d/yyyy H:mm:ss tt, where tt = AM or PM
- yyyy-MM-dd HH:mm:ss

The attribute event file can contain dimensional data and must also contain `ATTRIBUTE_NAME`, `DATE` or `TIMESTAMP`, `PRODUCED_QTY`, `TESTED_QTY`, and `FAILED_QTY` columns. If both the `DATE` and `TIMESTAMP` column exist, the `DATE` column is used and the `TIMESTAMP` column is not. The `DATE` column must use one of the following date formats:

- dd/MM/yyyy
- yyyy/MM/dd
- yyyy/M/d
- d/M/yyyy(

  **Note:** M/d/yyyy format is assumed if no day is higher than 12
  )
- dd.MM.yyyy
- MM-dd-yyyy
- M/d/yyyy
- yyyy-MM-dd

The variable parameter file can contain dimensional data as well as the following required and optional columns (not case-sensitive):

- QEWS_ACCEPT_LEVEL (required)
- QEWS_SIGMA (required)
- QEWS_TARGET (required)
- QEWS_UNACCEPT_LEVEL (required)
- QEWS_VARIABLE or QEWS_VARIABLE_NAME (required)
- QEWS_FALSE_ALARM_RATE (optional)
- QEWS_NUMBER_OF_POINTS (optional)
- QEWS_RUNTIME (optional)
- QEWS_TYPE_OF_CONTROL (optional)

The attribute parameter file can contain dimensional data as well as the following required and optional columns (not case sensitive):

- QEWS_ATTRIBUTE_NAME (required)
- QEWS_LAM0 (required)
- QEWS_LAM1 (required)
- QEWS_NUMBER_OF_VINTAGES (optional)
- QEWS_PROB (optional)
- QEWS_RUNDATE (optional)

**Important:**

- File names must not contain double pound symbols (##), less-than-greater-than symbols (<>), or square brackets ([]).
- Dimension column names must not contain special characters or spaces.
- Values in dimension columns must not contain colons (:), semicolons (;), or commas (,).
- Values in the ATTRIBUTE_NAME column must not begin or end with a pound symbol (#) and must not contain double pound symbols (##) or semicolons (;).
- The QEWS_RUNTIME column of parametric parameter must use the yyyy-MM-dd HH:mm:ss time format.
- The QEWS_RUNDATE column of attribute parameter must use the yyyy-MM-dd date format.
- The column headings ORG_ID, DEVICE_TYPE, DEVICE_ID, and EVENT_TYPE are used by IoT event files. These columns are always identified as dimensions.
- Files must not contain duplicate column names.
- In each file, date and time formats must be consistent, otherwise the file parser fails.
- During file upload, the file parser identifies file types by using the following rules. If the file contains a QEWS_ATTRIBUTE_NAME column, then it is an attribute parameter file. If the file contains a QEWS_VARIABLE_NAME or QEWS_VARIABLE column, then it is a variable parameter file. If the file contains an ATTRIBUTE_NAME column, then it is an attribute event file. Otherwise, it is a variable event file.

## Procedure

1. Select **My Data** > **Files** > **Browse for file**, select the file, and select **Open**.
2. In the Progress pane, view the status of the upload.
3. In the Uploaded Files section, view the files. A file cannot be deleted while it is being parsed by the system. When the parsing is done, hover over the tags in

each file card to see the object type, dimensions, and other information. (If the parsing fails, then an error message appears. Delete the file, correct the problem in the file, and upload again.)

4. To preview the data in the file, click `Preview`. The first 10 rows of the data are shown.

## Downloading data files

On the home page, you can download a data file that has been uploaded.

### Procedure

1. Select **My Data** > **Connections** > **Files**.
2. In the Uploaded Files section, view the files.
3. From the list of files, click the menu button of the file and click **Download**.

## Deleting data files

On the home page, you can delete a data file that was uploaded.

### About this task

When you delete a file, it does not delete the variables, parameters, or analysis results that relate to the file, but deletes the file from the system. When the status of a file is Parsing, Collecting & Parsing, or Collecting, you cannot delete the file.

### Procedure

1. Select **My Data** > **Connections** > **Files**.
2. In the Uploaded Files section, view the files.
3. From the list of files, select the menu of the file and select **Delete**.

## Viewing object types

After an event data file is uploaded, you can view the detected variable types and attribute types.

### Procedure

1. Select **My Data** > **Objects** > **Variables** or **My Data** > **Objects** > **Attributes**.
2. View the analyzed variables or attributes. If a variable or attribute has a missing parameter, an alert is shown: **Missing Parameters**.

## Rules for finding matched parameters in object analysis

When you set analysis parameters for an attribute type or a variable type, rules are used to find matched parameters. The rules in the following list are used in sequential order to find matched parameters in object analysis.

- To find matched parameters for the object, the system looks for dimension values that match. For example, if the variable *V1* had plantA and productA dimensions, the existing parameter records are searched for dimensions that are plantA and productA. If a variable *V1* is found that has the dimensions plantC and productA, or a variable *V1* is found with the dimensions plantA, or if a variable *V1* is found with the dimensions plantA, productA and operationA, the parameter is not considered a match because the dimensions do not match. If multiple parameter records are matched, the most recent record is used.

- If no parameters are found, parameters for the object without dimensions are searched. Any records that have dimensions are not considered. If multiple parameter records that do not have dimensions exist for the object, the most recent record is used.
- If no parameters are found in first and second step, parameters for the object with other dimensions are searched. The most recent valid record for the object is used, whether dimensions are specified or not.

## Setting analysis parameters for a variable type

To run an IBM Quality Early Warning System parametric analysis, you must set the parameters. A variable type analysis is for numeric data such as temperature, volume, or height.

### About this task

You can set the parameters by uploading a parameter file or by using the following procedure. Use the following table for parameter values.

*Table 1. Parameter data types and descriptions*

| Parameter name | Data type | Description and setting |
|---|---|---|
| **`Runtime`** | Time | The time point from when time is counted back for the number of points that are included in the analysis. The default value is the system time for when the analysis is run. The format must be `yyyy-MM-dd HH:mm:ss`. |
| **`Sigma`** | Double | The assumed standard deviation of the measurements. The value must be greater than zero. |
| **`Target`** | Double | The most desirable value for the center or mean measurement of the measured population. |
| **`Acceptable level`** | Double | The deviation from the mean of the measurements that is acceptable. Typically this level is close to the Target value and reflects the amount of movement for the population mean around the target. Where these is low process capability this level coincides with the Target, indicating that there is no movement for the population mean. |

*Table 1. Parameter data types and descriptions (continued)*

| Parameter name | Data type | Description and setting |
|---|---|---|
| **`Unacceptable level`** | Double | The deviation from the mean of the measurements that is unacceptable. You must have good detection capability for this level to work well. The unacceptable level must be further the Target than the acceptable level. Maintain a degree of separation between the acceptable and unacceptable levels of 0.2*Sigma. |
| **`Type of control`** | int | If Type_of_control = 1, then the control is one-sided. Only changes up or changes down are detected.<br>If Type_of_control = 2, then the control is two-sided: both types of deviation from the Target are detected.<br>For example: If Type_of_Control = 1 and Accept_Level < Unaccept_Level, then only increased changes of the process mean are flagged. If Type_of_Control = 1 and Accept_Level > Unaccept_Level, then only decreased changes of the process mean down are flagged.<br>If Type_of_Control = 2 then you specify either Accept_Level < Unaccept_Level or Accept_Level > Unaccept_Level, with the understanding that the acceptable and unacceptable levels of the two-sided procedure are positioned symmetrically around the target. |

*Table 1. Parameter data types and descriptions (continued)*

| Parameter name | Data type | Description and setting |
|---|---|---|
| `False alarm rate` | Double | The default value is 5000. The detection procedure produces a rate of false alarms of 1 per 5000 points when the population mean is located at the LAM0. Generally, it is a good policy for the false alarm rate be at least 50X the expected window size (number of points); in this case, the false flagging probability per analysis is about 0.02. |
| `Number of points` | int | Total number of points in the analysis. The default value is 100. |

### Procedure

1. Select **My Data** > **Objects** > **Variables** and select the variable that you want to set.
2. Select **Apply and Save**.

**Related information**:

Rules for finding matched parameters in object analysis
When you set analysis parameters for an attribute type or a variable type, rules are used to find matched parameters. The rules in the following list are used in sequential order to find matched parameters in object analysis.

## Setting analysis parameters for an attribute type

To run an IBM Quality Early Warning System attribute analysis, you first need to set the parameters. An attribute type analysis is for pass or fail data.

### About this task

You can set the parameters by uploading a parameter file or by using the following procedure. Use the following table for parameter values.

*Table 2. Parameter data types and descriptions*

| Parameter name | Data type | Description and setting |
|---|---|---|
| `QEWS_ATTRIBUTE_NAME` | String | The name of the attribute to be analyzed. This is a mandatory parameter. |
| `QEWS_LAM0` | Double | The acceptable failure rate. Failure rate equals failed quantity divided by tested quantity. The range of LAM0 is (0,1). This is a mandatory parameter. |

*Table 2. Parameter data types and descriptions (continued)*

| Parameter name | Data type | Description and setting |
|---|---|---|
| `QEWS_LAM1` | Double | The unacceptable failure rate. LAM1 must be greater than LAM0. The range of LAM1 is (0,1). This is a mandatory parameter. |
| `QEWS_PROB` | Double | The probability of not having a false alarm of failure rate. The range of PROB is (0,1). The default value is 0.99. |
| `QEWS_NUMBER_OF_VINTAGES` | Integer | The maximum number of vintages in the analysis. The default value is 100. |
| `QEWS_RUNDATE` | Date | The date from which time is counted back for the number of vintages that are included in the analysis. The format must be yyyy-MM-dd. The default value is the system date for when the analysis is run. |

### Procedure

1. Select **My Data** > **Objects** > **Attributes** and select the attribute that you want to set.
2. Click **Apply and Save**.

**Related information**:

Rules for finding matched parameters in object analysis
When you set analysis parameters for an attribute type or a variable type, rules are used to find matched parameters. The rules in the following list are used in sequential order to find matched parameters in object analysis.

## Selecting the objects for analysis

You can add each variable type or attribute type to a board in the board view and run an analysis by selecting dimensions.

### Procedure

1. From the board view, click **Add objects**.
2. Select a variable from the Variables tab or select an attribute from the Attributes tab.
3. Select the dimensions to run with the object (variable or attribute). You can select multiple dimensions to run at one time.
4. Click **Add**. The system displays the analysis state of the new object.

### Results

During analysis, all existing dimension value combinations are detected and analyses are run for each combination. You can select **Add All** or **Custom Select** to define which combinations to add to the board or group.

# Organizing objects

You use boards to organize and track your objects. Within the boards, you can organize the objects into groups. You can edit the objects from within the boards and groups.

## Boards

Boards are used to organize objects. You create, view, and edit boards in the My Boards section.

Each board card summarizes the status of the objects found in the board. There are four color blocks in each board card that indicate the quality risk of the objects in the board. The blocks also indicate the percentage of objects that are in each color block.

- Red - Has exceeded the notification threshold within the last 10 data points.
- Yellow - Within threshold but has data points that are nearing the alert threshold.
- Blue - Alerts were issued before the last 10 data points, but now returning to an acceptable condition.
- Gray - All data points are within threshold, or some points were out of threshold but recent points are within threshold.

For example, a red color block showing 39% means that 39% of the objects that are associated with the board exceeded the notification threshold within the last 10 data points.

### Creating a board

You can create a board and add objects to the board.

#### Procedure

1. From the home page, click the **Create New Board** button.
2. Type the name of the new board.
3. Click **Create**.
4. If you want to change the name of the board, click **Edit Name** and type the new name of the board.
5. Click **Save**.
6. Click **Add objects** to add an object to the board.

### Deleting boards

You can delete a board.

#### Procedure

1. Click the menu button of the board you want to delete.
2. Click **Delete**.
3. In the confirmation window, click **Delete**.

## Groups

You can view all the groups in a board by selecting the board. You can organize objects within the groups of a board.

In a board, select **New Group** to create a group. You can edit the name of the group at any time. To edit the name of the group, click the current name of the group, edit the name, and click the check mark. You can also go to the menu of the group and select **Edit Name** to change the name of the group.

You add objects to the groups as a way to organize your objects. To add any number of objects to the groups, go to the menu of the group and select **Add Objects**. After you add an object to a group, you can view the quality analysis of the object.

If you no longer want a group, you can delete it. If the group contained objects, the objects are still in the Your Objects section. Go to the menu of the group and click **Delete** to delete the group.

## Filtering boards

You can use the Quality Early Warning System alert level and all objects as filter criteria.

### About this task

After saving, the filter tags will show.

### Procedure

1. From the Board view, click **Filter**.
2. Select the filter criteria from the Alert type and Object Type dialog boxes.
3. Click **Save**. The filter criteria shows.

## Editing the name of a group

You can change the name of a group in different ways.

### Procedure

1. To open the group in edit mode, from the Groups view, click the name of the group you want to change.
2. Type the new name of the group and click the check mark icon.

## Adding an object to a group

After analyzing new objects, you can add all the analyzed objects to a group or make a selection of the objects.

### About this task

If you choose to add a selection of objects, a dialog box guides you to make the selection.

### Procedure

1. In the Groups view, click **Add Objects**.
2. Select the group to which you want to add the objects.
3. Select objects. You can select **Add All** or **Custom Select** to further define which combinations to add to the board or group.
   - To select all objects, click **Add All**.
   - To make a selection of objects, click **Custom Select**. Do all of the following steps.
4. A list of ungrouped objects opens. Select objects to add to the group.
5. Click **Add**.

### Moving a group from one board to another

A group can be moved from a board to another board, or to a new position in its present board.

#### About this task

If the target board contains the same group name as the group name to be moved, then you can merge with, or replace, the group in the target board.

#### Procedure

1. In the groups view, click the group menu and click **Move**.
2. Select the target board.
3. Select the position in the target board and click **Save**.
4. If a group exists with the same group name, you see a dialog box to make the selection:
   - **Merge** merges the group data in the target board with the data in the moved group.
   - **Replace** overwrites the group in the target board with the data in the moved group.
   - **Save anyway** automatically renames the moved group. The data of both groups is separately retained.

### Deleting a group from a board

A group can be deleted from a board.

#### Procedure

1. In the groups view, click the group menu and click **Delete**.
2. In the confirmation window click **Delete**.

### Moving an object from one group to another

An object can be moved from one group or board to another other group or board.

#### About this task

If the target board or group contains the same object name as the object name to be moved, decide whether you want to replace the object in the target.

You can drag an object or use the following steps.

#### Procedure

1. From the Object view, click the object menu and click **Move**.
2. Select the board, group, and position, then click **Save**.

### Deleting an object from a group

An object can be deleted from a group.

#### Procedure

1. From the Object view, click the object menu and click **Delete**.
2. From the delete confirmation window, click **Delete**.

## Variable analysis reports

You can see the report of a variable from a group.

Each analysis report has a color tag that gives information over the status of the variable. These are described in the following table:

*Table 3. Variable status colors and condition*

| Label | Color | Condition |
|---|---|---|
| Acceptable | Grey | All data points are within threshold or there are some points out of threshold but recent points has gone back within threshold. |
| Caution | Amber | Within threshold but has points that are verging to alert threshold. |
| Alert | Red | Has exceeded the notification threshold within the last 10 data points. |
| Alert but improving | Blue | Alerts issued before the last 10 data points, verging back to an acceptable condition. |

The report charts shows two charts:

**Variable Value Chart**
Shows graphically the variable value, the target value and the upper and lower unacceptable level values.

**Evidence Chart**
Shows graphically the trend over time for the number of data points. The default number is 100.

If the parameter for `Type_of_control` is set to 1 sided, then points outside the notification threshold, and in some cases even individual points within threshold for which the supplemental *Shewhart* test is violated, are marked with solid blue triangles and corresponds to the legend **Alert Lower** and **Alert Upper**.

If the parameter for the `Type_of_control` is set to 2 sided then points outside the upper notification threshold are marked with blue triangles and corresponds to the legend **Alert Upper**, and points outside the lower notification threshold violating points are marked with red triangles and corresponds to the legend **Alert Lower**.

Below the tag shows the title of the variable and the name of the predictive data model.

The time selector enables you to set the analysis date.

## Attribute analysis reports

You can see the report for each attribute in a group.

*Vintage* represents a group of products or work pieces that you want to analyze together to find quality issues. The input data is aggregated to the vintage level and the analysis is performed at the vintage level. In each analysis, vintage is identified by the `DATE` column (the same `ATTRIBUTE_NAME` and the same dimension values are used), so vintage is synonymous with *vintage date*. The `PRODUCED_QTY`, `TESTED_QTY`, and `FAILED_QTY` values reflect the vintage date.

The analysis identifies unacceptable levels of failure rate. Failure rate is calculated by `FAILED_QTY/TESTED_QTY` for the same vintage for the selected attribute and dimension value.

During analysis, the main algorithm is the *Weighted Pages* algorithm. Supplemental test #1 identifies unacceptably high numbers of failures in the last **`n_days_active`** calendar days (by default, **`n_days_active`** is 60). Supplemental test #2 is based on the last value of the Weighted Pages algorithm if the last vintage is within **`n_days_active`** calendar days. The test results are combined to establish a severity level that determines if an alarm should be triggered. The last good period is also determined to clearly identify a *bad regime* (a bad series of data) and a *good regime* (a good series of data).

The online article *Weighted Cumulative Sum Technique* by Emmanuel Yashchin provides more information about the Weighted Pages algorithm.

Each analysis report indicates the status of the analysis result, as described in the following table:

*Table 4. Failure rate status colors and condition*

| Color | Label | Description | Chart characteristics | Rules |
|---|---|---|---|---|
| Red | Alert | Unforgiven violations exist. It's time to take action. | The blue curve is above its threshold and data points have blue triangles, or a "+" or "x" marker appears at the last vintage. | `If (severity>=prob & forgiveness<9 & (last_alerting_vintage within 6 days or supalarm1 = 1 or (supalarm1 = 2 & forgiveness = 0)))` |
| Blue | Alert but improving | Recent forgiven violations exist, or unforgiven violations exist but the situation has improved. | The blue curve is above its threshold and data points have blue triangles, or a "W" marker appears at the last vintage. | `Else if (severity>=prob & ((forgiveness = 9 & last_alerting_vintage within 20 days) or (forgiveness <9 & supalarm1 = 0) or (1<forgiveness <9 & supalarm1 = 2)))` |
| Amber | Caution | No unforgiven violations exist, but recent or last vintages are near high severity. Keep watch over this. | A "W" marker appears at the last vintage. | `Else if (1-severity >= (1-prob) * fact_yellow & supalarm1 = 2)` |
| Grey / No color | Acceptable | No issues exist, or the issues are old and the situation has returned to normal. No actions are required. | There are no "+", "x", or "W" markers. There might be blue triangle alerts that have been forgiven. | `Else.` |

*Severity* is the maximum value of Weighted Pages and supplemental test #1 and #2. Severity is in the range (0,1). Severity 1 means that the degree of violation in the evidence trajectory is so strong that there is zero probability that the process is acceptable.

*Supalarm1* is determined by supplemental test #1: if supplemental test #1 severity is above *prob*, then it is 1; if supplemental test #1 severity is between *prob* and 1-(1-*prob*)**fact_yellow*, then it is 2; otherwise it is 0.

*Forgiveness* is the level of evidence that failure rate has returned to an acceptable level following a violation. Forgiveness is in the range (0,9), where 9 means the violation is forgiven and 0 means that there is no evidence of returning to an acceptable level.

`Fact_yellow` is set to 1.5.

The report contains the following charts:

**Failure Rate Chart**
The Failure Rate chart shows the failure rate values and the acceptable level of failure rate. The X-axis top scale represents the accumulated tested quantity. The X-axis bottom scale represents the sequence number of the vintage (from oldest to newest) counted from the attribute data.

**Evidence Chart**
The Evidence Chart shows the time trend for the number of vintages. If the supplemental test #1 alarm is triggered, the last point of the evidence curve appears as "+". If the supplemental test #2 alarm is triggered, the last point of the evidence curve appears as "x". If supplemental test #1 or #2 alarm is in warning status, the last point of the evidence curve appears as "W".

## Modifying a parameter and rerunning an analysis

You can edit a parameter from the analyzed object and rerun the anaysis.

### Procedure

1. From the object analysis report, click **View Parameter**.
2. The dialog box shows the current analysis parameters, click **Edit Parameters**.
3. Modify the values of the parameters that you need.
4. Click **Save and Run**.

# Notification

You can receive system notifications in the product.

The notification icon shows the number of notifications and lists the notifications in chronological order.

The notification types are:

**File uploaded**
The user uploaded a file.

**File deleted**
The user deleted a file.

**Variable analysis complete**
An analysis of a variable is complete.

**Attribute analysis complete**
An analysis of an attribute is complete.

**Variable types added**
Variables were added to the system.

**Attribute added**
Attributes were added to the system.

**Missing parameters**
A number of variable types or attributes are missing parameter settings.

**Errors found**
Shows the cause of the error and indicates any missing parameters from an object, errors that were caused by the analysis of an object, and any system failure to run an analysis.

# Chapter 5. Integrating with Watson IoT Platform

You can integrate IBM Prescriptive Quality for Manufacturing Private with IBM Watson® IoT Platform to collect product quality data from connected devices and perform quality early warning system (QEWS) analysis.

Before you create an integration, make sure that your machine can access to internet. After you create an integration, all connected devices are detected and data collection begins. Only device data that is in simple JSON format is collected. Device data is stored as event files in CSV format. The files are automatically parsed to identify the variables, attributes, and dimensions. If a variable and dimension combination exists on a board, analysis occurs, and the results are displayed on the board.

One or two event files are created each day for data from each organization, device type, and data schema. Files are named *<orgId>_<deviceType>_yyMMddHHmmss.SSS*.csv, where *yyMMddHHmmss.SSS* is the time that the file was created. When a new file is created, the file is parsed to identify the variables, attributes, and dimensions. At the end of each day, or when an integration is disconnected or deleted, the event files are closed and fully parsed.

The event files from Watson IoT Platform integrations are treated like other uploaded files during file parsing. The same rules for identifying file types, time stamps, variables, attributes, and dimensions are applied. If a file contains objects and dimensions, they are included in analysis regardless of whether the file is a Watson IoT Platform event file or an uploaded event file.

## Watson IoT Platform event files

When you integrate Prescriptive Quality for Manufacturing Private with Watson IoT Platform, device data is stored as event files in CSV format. One or two event files are created each day for data from each organization, device type, and data schema. At the end of each day, or when an integration is disconnected or deleted, the event files are closed and fully parsed.

Files are named `<orgId>_<deviceType>_yyMMddHHmmss.SSS.csv`, where *yyMMddHHmmss.SSS* is the time that the file was created.

The event files from Watson IoT Platform integrations are treated like other uploaded files during file parsing. The same rules for identifying file types, time stamps, variables, attributes, and dimensions are applied. If a file contains objects and dimensions, they are included in analysis regardless of whether the file is a Watson IoT Platform event file or an uploaded event file.

## Creating integrations

You can integrate Prescriptive Quality for Manufacturing Private with Watson IoT Platform to collect product quality data from devices that are connected to Watson IoT Platform and perform quality early warning system analysis.

### Before you begin

Before you integrate Prescriptive Quality for Manufacturing Private with Watson IoT Platform, you must complete the following tasks:

- Configure an organization in Watson IoT Platform. The organization must contain at least one device type and one device.
- Copy the strings for the organization ID, API key, and authentication token into a text file.

### Procedure

1. Select **My Data** > **Connections** > **IoT** and then select **Add Integration**.
2. Enter the organization ID, API key, and authentication token, and then select **Connect**.
3. Select the devices from which you want to receive data.
4. Save the integration.

### What to do next

After you create an integration with a Watson IoT Platform organization, you can connect or disconnect the integration to start or stop the collection of data from the devices in that organization. Select **My Data** > **Connections** > **IoT**, select the menu of the integration, and then connect or disconnect the integration.

## Editing an integration

After you create an integration with a Watson IoT Platform organization, you can edit the integration, for example, to change the devices from which you collect data.

### Procedure

1. Select **My Data** > **Connections** > **IoT**.
2. Select the menu of the integration that you want to edit and then select **Disconnect**.
3. Edit the integration.
4. Save the integration.
5. Reconnect the integration.

# Chapter 6. Application programming interface

The application programming interface provides actions such as enabling or triggering an IBM Quality Early Warning System analysis on a variable or attribute, setting or removing an analysis interval, and getting the analysis results.

## Enable QEWS analysis on a variable independently

Enables the IBM Quality Early Warning System analysis for a variable's data independently from your other data on IBM Prescriptive Quality for Manufacturing Private. The event data itself and the parameters must be provided directly in the request. The analysis result is included in the response directly.

**URL**

/ibm/pq/service/QEWSVariable

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`VARIABLE_NAME: <String>`. Contains the name of the variable to be analyzed.

`EVENT_TIME: <JSONArray>`. Contains timestamp values in the sequence that you want to analyze data. Accepted format is `yyyy-MM-dd HH:mm:ss`.

`VARIABLE_VALUE: <JSONArray>`. Contains numeric values in the same sequence of timestamps.

`RUNTIME: <String>`. Runtime. Mandatory. The time point from when time is counted back for the number of points that are included in the analysis. Accepted format is `yyyy-MM-dd HH:mm:ss`.

`PARAMETER: <JSONObject>`. Contains parameters for doing the QEWS variable analysis, including the following items:

- `QEWS_TARGET: <Double>`. Mandatory.
- `QEWS_SIGMA: <Double>`. Mandatory.
- `QEWS_ACCEPT_LEVEL: <Double>`. Mandatory.
- `QEWS_UNACCEPT_LEVEL: <Double>`. Mandatory.
- `QEWS_TYPE_OF_CONTROL: <Integer>`. Optional. Default is 2.
- `QEWS_FALSE_ALARM_RATE: <Double>`. Optional. Default is 5000.
- `QEWS_NUMBER_OF_POINTS: <Integer>`. Optional. Default is 100.

## Sample body

```
[{
 "Username":"<IBMid>",
 "VARIABLE_NAME":"Height",
 "EVENT_TIME":["2017-06-01 00:00:00","2017-06-01 00:10:00"],
 "VARIABLE_VALUE":["10","11"],
 "RUNTIME":"2017-06-02 00:00:00",
 "PARAMETER":{
  "QEWS_TARGET":"10",
  "QEWS_SIGMA":"1",
  "QEWS_ACCEPT_LEVEL":"11",
  "QEWS_UNACCEPT_LEVEL":"12",
  "QEWS_TYPE_OF_CONTROL":"2",
  "QEWS_FALSE_ALARM_RATE":"5000",
  "QEWS_NUMBER_OF_POINTS":"100"
 }
}]
```

## Success response

201 Created

```
[
    {
        "result": {
            "kpi": {
                "violation_type": [
                    "0.0",
                    "0.0"
                ],
                "scheme1_val": [
                    "0.0",
                    "0.0"
                ],
                "scheme2_val": [
                    "0.0",
                    "0.0"
                ],
                "variable_val": [
                    "10.0",
                    "11.0"
                ],
                "variable_timestamp": [
                    "2017-06-01 00:00:00",
                    "2017-06-01 00:10:00"
                ]
            },
            "profile": {
                "accept_level": "11.0",
                "false_alarm_rate": "5000",
                "lastbadptvalue": "0.0",
                "runtime": "2017-06-02 00:00:00",
                "severity": "0.0",
                "forgiveness": "0.0",
                "lastbadptindex": "-1",
                "rc": "0.0",
                "rcun": "0.0",
                "threshold_lower": "6.70007",
                "num_points": "100",
                "result_record": "2",
                "threshold_upper": "6.70007",
                "sigma": "1.0",
                "unaccept_level": "12.0",
                "variable_color": "1",
                "target": "10.0",
                "unaccept_factor_sigma": "1.5",
                "type_of_control": "2"
            }
        },
        "warning_message": ""
    }
]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "error_message": "<error messages content. >"
    }
]
```

**Response items**

`kpi`: `<JSONObject>`. Contains the first set of QEWS analysis results, including the violation type and evidence value of each analyzed event data.

`violation_type`: `<JSONArray>`. Contains the violation type of each analyzed event data.

- `0` means there is no violation for the data point.
- `>0` means there is an upper direction violation. The event data value is too big.
- `<0` means there is a lower direction violation. The event data value is too small.

`scheme1_val`: `<JSONArray>`. Contains the first set of evidence values of each analyzed event data.

- If the analysis is on both the upper and lower directions, `scheme1_val` contains the evidence value of the upper direction.
- If the analysis is on the upper direction only, `scheme1_val` contains the evidence value of the upper direction
- If the analysis is on the lower direction only, `scheme1_val` contains the evidence value of the lower direction

The value is always non-negative, so when it contains the evidence value of the lower direction, you might use `0-scheme1_val` to draw on the evidence chart.

`scheme2_val`: `<JSONArray>`. Contains the second set of evidence values of each analyzed event data.

- If the analysis is on both the upper and lower directions, `scheme2_val` contains the evidence value of the lower direction.
- If the analysis is on the upper direction only, `scheme2_val` contains empty values.
- If the analysis is on the lower direction only, `scheme2_val` contains empty values.

The value is always non-negative, so when it contains the evidence value of lower direction, you might use `0-scheme2_val` to draw on the evidence chart.

`variable_val`: `<JSONArray>`. Contains the variable value of each analyzed event data.

`variable_timestamp`: `<JSONArray>`. Contains the timestamps of each analyzed event data.

profile: <JSONObject>. Contains the second set of QEWS analysis results, including the values of the QEWS parameters in the analysis, and the QEWS calculated indicators such as severity and forgiveness.

accept_level: <Double>. From the input parameter QEWS_ACCEPT_LEVEL.

false_alarm_rate: <Integer>. From the input parameter QEWS_FALSE_ALARM_RATE. If the value does not exist in the input parameter, the default value is used.

lastbadptvalue: <Double>. Last bad point value. The value of the last data point that is consistent with unacceptable behavior.

runtime: <String>. From the input parameter RUNTIME.

severity: <Double>. Degree of violation of acceptable conditions that are detected during analysis. This is a base-10 logarithm of the "conformance to acceptable process level" test p-value.

forgiveness: <Integer>. Degree of forgiveness that is observed within the last acceptable period of the data. The value ranges from 0 to 9.

- 1 means that the degree of forgiveness is very low.
- 9 is achieved when the forgiveness criteria are satisfied.

lastbadptindex: <Integer>. Last bad point index. The index of the last data point that is consistent with unacceptable behavior.

rc: <Integer>. Return code.

- Zero means a normal run.
- Positive values correspond to error conditions.
- Negative values give the minutes that elapsed since the last point that was consistent with the unacceptable process level.

rcun: <Integer>. Second return code. The number of minutes that elapsed between the timestamp of the last bad point and the timestamp of the last data point.

threshold_lower: <Double or NaN>. The QEWS calculated threshold for the lower direction evidence values. If the analysis is on the upper direction only, the value is "not a number" (NaN).

num_points: <Integer>. From the input parameter QEWS_NUMBER_OF_POINTS. If the value does not exist in the input parameter, the default value is used.

result_record: <Integer>. The number of data points that are analyzed.

threshold_upper: <Double or NaN>. The QEWS calculated threshold for the upper direction evidence values. If the analysis is on the lower direction only, the value is "not a number" (NaN).

sigma: <Double>. From the input parameter QEWS_SIGMA.

unaccept_level: <Double>. From the input parameter QEWS_UNACCEPT_LEVEL.

variable_color: <Integer>. Each number presents a status and color that is classified by QEWS.

- 1 means "Acceptable" and "Grey".
- 2 means "Caution" and "Amber" (Yellow).
- 3 means "Alert" and "Red".
- 4 means "Alert but improving" and "Blue".

target: <Double>. From the input parameter QEWS_TARGET.

unaccept_factor_sigma: <Double>. An input parameter of QEWS. Default value is used in all cases.

type_of_control: <Integer>. From the input parameter QEWS_TYPE_OF_CONTROL. If the value does not exist in the input parameter, the default value is used.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>","VARIABLE_NAME":"Height",
"EVENT_TIME":["2017-06-01 00:00:00","2017-06-01 00:10:00"],
"VARIABLE_VALUE":["10","11"],"RUNTIME":"2017-06-02 00:00:00",
"PARAMETER":{"QEWS_TARGET":"10","QEWS_SIGMA":"1","QEWS_ACCEPT_LEVEL":"11",
"QEWS_UNACCEPT_LEVEL":"12","QEWS_TYPE_OF_CONTROL":"2",
"QEWS_FALSE_ALARM_RATE":"5000","QEWS_NUMBER_OF_POINTS":"100"}}]'
-v "https://<serverhost>/ibm/pq/service/QEWSVariable"
```

**Notes**

This action is designed for normal-sized data processing. Do not exceed 5000 event data records.

The timestamps of the event data must be sorted by the right time sequence. Our service does not sort the data by timestamps. If there are two events with the same timestamp, our service still treats them as two data points and analyzes them with the sequence provided in your data.

## Enable QEWS analysis on an attribute independently

Enables the IBM Quality Early Warning System analysis for an attribute's data independently from your other data on IBM Prescriptive Quality for Manufacturing Private. The event data itself and the parameters must be provided directly in the request. The analysis result is included in the response directly.

**URL**

/ibm/pq/service/QEWSAttribute

**Method**

The request type POST

**URL parameters**

None

**Headers**

Content-Type: application/json

APIKEY: <encrypted key>. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

Username: <String>. Used to identify tenant and user. Mandatory.

ATTRIBUTE_NAME: <String>. Contains the name of the attribute to be analyzed.

VINTAGE_DATE: <JSONArray>. Contains the date values in the sequence that you want to analyze data. Accepted format is yyyy-MM-dd.

PRODUCED_QTY: <JSONArray>. Contains the quantity of produced units in the same sequence of vintages.

TESTED_QTY: <JSONArray>. Contains the quantity of tested units in the same sequence of vintages.

FAILED_QTY: <JSONArray>. Contains the quantity of failed units in the same sequence of vintages.

RUNDATE: <String>. Rundate. Mandatory. The date from when time is counted back for the number of vintages that are included in the analysis. Accepted format is yyyy-MM-dd.

PARAMETER: <JSONObject>. Contains parameters for doing the QEWS attribute analysis, including the following items:

- QEWS_LAM0: <Double>. The acceptable failure rate. Mandatory. The range of LAM0 is (0,1).
- QEWS_LAM1: <Double>. The unacceptable failure rate. Mandatory. The range of LAM1 is (0,1). LAM1 must be greater than LAM0.
- QEWS_PROB: <Double>. The probability of not having a false alarm of the failure rate. Optional. Default is 0.99. The range of PROB is (0,1).
- QEWS_NUMBER_OF_VINTAGES: <Integer>. The maximum number of vintages in the analysis. Optional. Default is 100.

**Sample body**

```
[{"Username":"<IBMid>","ATTRIBUTE_NAME":"P1","VINTAGE_DATE":["2017-06-01",
"2017-06-02"],"PRODUCED_QTY":["100","110"],"TESTED_QTY":["90","100"],
"FAILED_QTY":["1","2"],"RUNDATE":"2017-06-06","PARAMETER":{"QEWS_LAM0":
"0.01","QEWS_LAM1":"0.02","QEWS_PROB":"0.99","QEWS_NUMBER_OF_VINTAGES":
"100"}}]
```

**Success response**

201 Created

```
[
    {
        "result": {
            "kpi": {
                "vintage_no": [
                    "1",
                    "2"
                ],
                "failed_qty": [
                    "1.0",
                    "2.0"
                ],
                "violation": [
                    "false",
                    "false"
                ],
                "failure_rate": [
                    "0.011111111111111112",
                    "0.02"
                ],
                "vintage_date": [
                    "2017-06-01",
                    "2017-06-02"
                ],
                "failure_rate_vs_produced": [
                    "0.01",
                    "0.01818181818181818"
                ],
                "pct_failed_cumulate": [
                    "33.333333333333336",
                    "100.0"
                ],
                "produced_qty": [
                    "100.0",
                    "110.0"
                ],
                "cusum": [
                    "0.0",
                    "0.5573049591110358"
                ],
                "pct_tested_cumulate": [
                    "47.368421052631575",
                    "100.0"
                ],
                "tested_qty": [
                    "90.0",
                    "100.0"
                ],
                "pct_failed": [
                    "33.33333333333333",
                    "66.66666666666666"
                ]
            },
            "profile": {
                "sup_alarm1_flag": "0",
                "qews_number_of_vintages": "100",
                "severity": "0.571",
                "forgiveness": "0",
                "k": "0.014426950408889642",
                "last_bad_point": "",
                "rcun": "0.0",
                "threshold": "3.2588794223109683",
                "qews_attribute_name": "P1",
                "qews_rundate": "2017-06-06",
                "qews_lam0": "0.01",
                "severity_sup_alarm1": "0.0",
                "sup_alarm2_flag": "0",
                "qews_lam1": "0.02",
                "qews_prob": "0.99",
                "attribute_color": "1",
                "severity_sup_alarm2": "0.0"
            }
        },
        "warning_message": ""
    }
]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "error_message": "<error messages content. >"
    }
]
```

**Response items**

`kpi`: `<JSONObject>`. Contains the first set of QEWS analysis results, including the failure rate, violation flag, evidence value, and other statistics of each analyzed vintage.

`vintage_no`: `<JSONArray>`. Contains the vintage index of each analyzed vintage.

failed_qty: <JSONArray>. Contains the failed quantity of each analyzed vintage. This should be the same as the input data.

violation: <JSONArray>. Contains the violation flag of each analyzed vintage.

failure_rate: <JSONArray>. Contains the failure rate of each analyzed event data. The failure rate is calculated by the failed_qty/tested_qty for the same vintage.

vintage_date: <JSONArray>. Contains the date of each analyzed vintage.

failure_rate_vs_produced: <JSONArray>. Contains the failure rate versus produced quantity, which is calculated by the failed_qty/produced_qty for the same vintage.

pct_failed_cumulate: <JSONArray>. Contains the cumulated failed percentage of all failed units for each analyzed vintage.

produced_qty: <JSONArray>. Contains the produced quantity of each analyzed vintage. This should be the same as the input data.

cusum: <JSONArray>. Contains the evidence value of each analyzed vintage.

pct_tested_cumulate: <JSONArray>. Contains the cumulated tested percentage of all tested units for each analyzed vintage.

tested_qty: <JSONArray>. Contains the tested quantity of each analyzed vintage. This should be the same as the input data.

pct_failed: <JSONArray>. Contains the failed percentage of all failed units for each analyzed vintage.

profile: <JSONObject>. Contains the second set of QEWS analysis results, including the values of the QEWS parameters in the analysis, and the QEWS calculated indicators such as severity and forgiveness.

sup_alarm1_flag: <Integer>. Supplemental alarm 1 flag.

- 0 means there is no alarm.
- 1 means there is an alert.
- 2 means there is a warning.

qews_number_of_vintages: <Integer>. From the input parameter QEWS_NUMBER_OF_VINTAGES. If the value does not exist in the input parameter, the default value is used.

severity: <Double>. Degree of evidence that the process is unacceptable, in terms of a severity index that varies between 0 (low severity) and 1 (high severity).

forgiveness: <Integer>. Degree of forgiveness that is observed within the last acceptable period of the data. The value ranges from 0 to 9.

- 1 means that the degree of forgiveness is very low.
- 9 is achieved when the forgiveness criteria are satisfied.

k: <Double>. The value that is used to calculate the evidence curve.

last_bad_point: <Integer>. Last bad point index. The index of the last data point that is consistent with unacceptable behavior.

rcun: <Integer>. Return code.

- Zero means a normal run.
- Positive values correspond to error conditions.

- Negative values give the following results:
  - The integer part means the last condition that is consistent with unacceptable process behavior was observed (-RC) days ago.
  - The second decimal digit means the result of the first supplemental alert.
  - The third decimal digit means the result of the second supplemental alert.
  - The fifth decimal digit means the forgiveness value.

threshold: <Double>. The QEWS calculated threshold for the evidence values.

qews_attribute_name: <String>. The attribute name.

qews_rundate: <String>. From the input parameter RUNDATE.

qews_lam0: <Double>. From the input parameter QEWS_LAM0.

severity_sup_alarm1: <Double>. Supplemental alarm 1 severity. The range is between 0 (low severity) and 1 (high severity).

sup_alarm2_flag: <Integer>. Supplemental alarm 2 flag.

- 0 means there is no alarm.
- 1 means there is an alert.
- 2 means there is a warning.

qews_lam1: <Double>. From the input parameter QEWS_LAM1.

qews_prob: <Double>. From the input parameter QEWS_PROB. If the value does not exist in the input parameter, the default value is used.

attribute_color: <Integer>. Each number presents a status and color that is classified by QEWS.

- 1 means "Acceptable" and "Grey".
- 2 means "Caution" and "Amber" (Yellow).
- 3 means "Alert" and "Red".
- 4 means "Alert but improving" and "Blue".

severity_sup_alarm2: <Double>. Supplemental alarm 2 severity. The range is between 0 (low severity) and 1 (high severity).

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data ' [{"Username":"<IBMid>","ATTRIBUTE_NAME":"P1",
"VINTAGE_DATE":["2017-06-01","2017-06-02"],"PRODUCED_QTY":["100","110"],
"TESTED_QTY":["90","100"],"FAILED_QTY":["1","2"],"RUNDATE":"2017-06-06",
"PARAMETER":{"QEWS_LAM0":"0.01","QEWS_LAM1":"0.02","QEWS_PROB":"0.99",
"QEWS_NUMBER_OF_VINTAGES":"100"}}]'
-v "https://<serverhost>/ibm/pq/service/QEWSAttribute"
```

**Notes**

This action is designed for normal-sized data processing. Do not exceed 5000 event data records.

The dates of the event data must be sorted by the correct time sequence. Our service does not sort the data by dates. If there are two events with the same date, our service still treats them as two data points and analyzes them with the sequence provided in your data.

## Set analysis interval to a variable or attribute

Sets the periodic analysis interval to a variable or an attribute.

**URL**

/ibm/pq/service/setAnalysisInterval

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`variable_name: <String>`. Contains the name of the variable to be set.

`attribute_name: <String>`. Contains the name of the attribute to be set. Make sure no `variable_name` parameter exists when you want to set the attribute. Otherwise, the `variable_name` is used.

`dim_name: <String>`. Contains the dimension names. If there are multiple dimensions, separate each dimension name by a comma.

`dim_value: <String>`. Contains the dimension values. If there are multiple dimensions, separate each dimension name by a comma, and use the same sequence as `dim_name`.

`interval: <Integer>`. Contains the seconds value of the interval you want to set. Accepted values are `300` (5 minutes), `3600` (1 hour), and `86400` (1 day) only.

**Sample body**

```
[{"Username":"<IBMid>","variable_name":"Height","dim_name":"Plant,Product",
"dim_value":"Plant1,Product1","interval":"3600"}]
```

**Success response**

201 Created

```
[
    {
        "error_message":"",
        "result":"succeeded"
    }
]
```

If there is an error, then the following response content is returned:

```
[
    {
        "error_message":"<error messages content. >",
        "result":"failed"
    }
]
```

**Response items**

`result`: <String>. A message about whether the result succeeded or failed.

`error_message`: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data ' [{"Username":"<IBMid>","variable_name":"Height",
"dim_name":"Plant,Product","dim_value":"Plant1,Product1",
"interval":"3600"}]'
-v "https://<serverhost>/ibm/pq/service/setAnalysisInterval"
```

## Remove analysis interval from a variable or attribute

Removes the periodic analysis interval from a variable or an attribute.

**URL**

/ibm/pq/service/removeAnalysisInterval

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username`: <String>. Used to identify tenant and user. Mandatory.

`variable_name`: <String>. Contains the name of the variable to be removed.

`attribute_name`: <String>. Contains the name of the attribute to be removed. Make sure no `variable_name` parameter exists when you want to remove the attribute. Otherwise, the `variable_name` is used.

`dim_name`: <String>. Contains the dimension names. If there are multiple dimensions, separate each dimension name by a comma.

`dim_value`: <String>. Contains the dimension values. If there are multiple dimensions, separate each dimension name by a comma, and use the same sequence as `dim_name`.

**Sample body**

```
[{"Username":"<IBMid>","variable_name":"Height","dim_name":"Plant,Product",
"dim_value":"Plant1,Product1"}}]
```

**Success response**

201 Created

```
[
    {
        "error_message":"",
        "result":"succeeded"
    }
]
```

If there is an error, then the following response content is returned:

```
[
    {
        "error_message":"<error messages content. >",
        "result":"failed"
    }
]
```

**Response items**

`result`: <String>. A message about whether the result succeeded or failed.

`error_message`: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data ' [{"Username":"<IBMid>","variable_name":"Height",
"dim_name":"Plant,Product","dim_value":"Plant1,Product1"}]'
-v "https://<serverhost>/ibm/pq/service/removeAnalysisInterval"
```

## Upload files

Enables the upload of one or more files to IBM Prescriptive Quality for Manufacturing Private. The upload file information must be provided directly in the request. The upload result and `file_id` information are included in the response directly.

**URL**

/ibm/pq/service/uploadFile

**Method**

The request type POST

**URL parameters**

`Username`: <String>. Used to identify tenant and user.

**Headers**

`Content-Type: application/json`

`APIKEY`: <encrypted key>. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

None

**Success response**

```
{"file_id":{"<userfilename>":"<file_id>"},
"error_message":{"<userfilename>":"<error message content. >"}}
```

**Response items**

`file_id`: <JSONObject>. Contains the file identification information for each uploaded file.

error_message: <JSONObject>. Contains the error message for each file that failed to upload.

**Sample call**

```
curl -k -X POST -H "APIKEY:<encrypted key>"
-H "Accept-Language:en-US,en;q=0.5" -H "Content-Type:multipart/form-data"
-H "Connection:keep-alive" -F "data=@<userfilename>"
"https://<serverhost>/ibm/pq/service/uploadFile?Username=<username>"
```

## Trigger QEWS analysis for variables

Triggers the IBM Quality Early Warning System analysis for variables. The analysis uses the event data and parameter data on IBM Prescriptive Quality for Manufacturing Private.

**URL**

/ibm/pq/service/triggerVariableAnalysis

**Method**

The request type POST

**URL parameters**

None

**Headers**

Content-Type: application/json

APIKEY: <encrypted key>. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

Username: <String>. Used to identify tenant and user. Mandatory.

variables: <JSONArray>. Contains all of the variables for doing the QEWS variable analysis, including the following items.

variable_name: <String>. Contains the name of the variable to be analyzed.

dimensions: <JSONArray>. Contains the dimension names. The combination of the dimension names is used to select the data for the analysis. Optional.

dimension_values: <JSONArray>. Contains the dimension values. The combination of the dimension values is used to select the data for the analysis. Use the same sequence as dimensions. Optional.

runtime: <String>. Timestamp. Mandatory. The time point from when time is counted back for the number of points that are included in the analysis. Accepted format is yyyy-MM-dd HH:mm:ss.

PARAMETER: <JSONObject>. Contains parameters for doing the QEWS variable analysis. Optional. If you do not specify any parameters, all parameters are taken from the parameter file that you uploaded or set from the user interface in your history. Including the following items:

- QEWS_TARGET: <Double>. Mandatory.
- QEWS_SIGMA: <Double>. Mandatory.
- QEWS_ACCEPT_LEVEL: <Double>. Mandatory.

- QEWS_UNACCEPT_LEVEL: <Double>. Mandatory.
- QEWS_TYPE_OF_CONTROL: <Integer>. Optional. Default is 2.
- QEWS_FALSE_ALARM_RATE: <Double>. Optional. Default is 5000.
- QEWS_NUMBER_OF_POINTS: <Integer>. Optional. Default is 100.

**Sample body**

```
[{"Username":"<IBMid>","variables":[{"variable_name":"Height",
"dimensions":["Plant","Product"],"dimension_values":["Plant1","Product1"],
"runtime":"2017-06-02 00:00:00","PARAMETER":{"QEWS_TARGET":"10",
"QEWS_SIGMA":"1","QEWS_ACCEPT_LEVEL":"11","QEWS_UNACCEPT_LEVEL":"12",
"QEWS_TYPE_OF_CONTROL":"2","QEWS_FALSE_ALARM_RATE":"5000",
"QEWS_NUMBER_OF_POINTS":"100"}}]}]
```

**Success response**

201 Created

```
[
    {
        "variables":
        [
            {
                "variable_name":"Height",
                "dimensions":["Plant","Product"],
                "dimension_values":["Plant1","Product1"],
                "runtime":"2017-06-02 00:00:00",
                "warning_message":"",
                "error_message":"",
                "result":"succeeded"
            }
        ],
        "error_message":""
    }
]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "variables":
        [
            {
                "variable_name":"Height",
                "dimensions":["Plant","Product"],
                "dimension_values":["Plant1","Product1"],
                "error_message":"<error messages content. >",
                "result":"failed"
            }
        ],
        "error_message":"<error messages content. >"
    }
]
```

**Response items**

variables: <JSONArray>.

variable_name: <String>.

dimensions: <JSONArray>.

dimension_values: <JSONArray>.

runtime: <String>.

result: <String>. A message about whether the result succeeded or failed.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>",
"variables":[{"variable_name":"Height","dimensions":["Plant","Product"],
"dimension_values":["Plant1","Product1"],"runtime":"2017-06-02 00:00:00",
"PARAMETER":{"QEWS_TARGET":"10","QEWS_SIGMA":"1","QEWS_ACCEPT_LEVEL":"11",
"QEWS_UNACCEPT_LEVEL":"12","QEWS_TYPE_OF_CONTROL":"2",
"QEWS_FALSE_ALARM_RATE":"5000","QEWS_NUMBER_OF_POINTS":"100"}}]}]'
-v "https://<serverhost>/ibm/pq/service/triggerVariableAnalysis"
```

**Notes**

If you get a successful response, then you triggered the analysis successfully. The backend analysis might run for some time. However, a successful trigger does not guarantee a successful analysis.

## Trigger QEWS analysis for attributes

Triggers the IBM Quality Early Warning System analysis for attributes. The analysis uses the event data and parameter data on IBM Prescriptive Quality for Manufacturing Private.

**URL**

/ibm/pq/service/triggerAttributeAnalysis

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`attributes: <JSONArray>`. Contains all of the attributes for doing the QEWS attribute analysis, including the following items.

`attribute_name: <String>`. Contains the name of the attribute to be analyzed.

`dimensions: <JSONArray>`. Contains the dimension names. The combination of the dimension names is used to select the data for the analysis. Optional.

`dimension_values: <JSONArray>`. Contains the dimension values. The combination of the dimension values is used to select the data for the analysis. Use the same sequence as `dimensions`. Optional.

`rundate: <String>`. Rundate. Mandatory. The date from when time is counted back for the number of vintages that are included in the analysis. Accepted format is `yyyy-MM-dd`.

PARAMETER: <JSONObject>. Contains parameters for doing the QEWS attribute analysis. Optional. If you do not specify any parameters, all parameters are taken from the parameter file that you uploaded or set from the user interface in your history. Including the following items:

- QEWS_LAM0: <Double>. The acceptable failure rate. Mandatory. The range of LAM0 is (0,1).
- QEWS_LAM1: <Double>. The unacceptable failure rate. Mandatory. The range of LAM1 is (0,1). LAM1 must be greater than LAM0.
- QEWS_PROB: <Double>. The probability of not having a false alarm of the failure rate. Optional. Default is 0.99. The range of PROB is (0,1).
- QEWS_NUMBER_OF_VINTAGES: <Integer>. The maximum number of vintages in the analysis. Optional. Default is 100.

**Sample body**

```
[{"Username":"<IBMid>","attributes":[{"attribute_name":"PRODUCT400",
"dimensions":["MODEL","PROCESS"],"dimension_values":["MODEL1","P1"],
"rundate":"2017-06-06","PARAMETER":{"QEWS_LAM0":"0.01","QEWS_LAM1":"0.02",
"QEWS_PROB":"0.99","QEWS_NUMBER_OF_VINTAGES":"100"}}]}]
```

**Success response**

201 Created

```
[
    {
        "attributes":
        [
            {
                "attribute_name":"PRODUCT400",
                "dimensions":["MODEL","PROCESS"],
                "dimension_values":["MODEL1","P1"],
                "rundate":"2017-06-06",
                "warning_message":"",
                "error_message":"",
                "result":"succeeded"
            }
        ],
        "error_message":""
    }
]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "attributes":
        [
            {
                "attribute_name":"PRODUCT400",
                "dimensions":["MODEL","PROCESS"],
                "dimension_values":["MODEL1","P1"],
                "error_message":"<error messages content. >",
                "result":"failed"
            }
        ],
        "error_message":"<error messages content. >"
    }
]
```

**Response items**

attributes: <JSONArray>.

attribute_name: <String>.

dimensions: <JSONArray>.

dimension_values: <JSONArray>.

rundate: <String>.

result: <String>. A message about whether the result succeeded or failed.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>",
"attributes":[{"attribute_name":"PRODUCT400",
"dimensions":["MODEL","PROCESS"],"dimension_values":["MODEL1","P1"],
"rundate":"2017-06-06","PARAMETER":{"QEWS_LAM0":"0.01","QEWS_LAM1":"0.02",
"QEWS_PROB":"0.99","QEWS_NUMBER_OF_VINTAGES":"100"}}]}]'
-v https://<serverhost>/ibm/pq/service/triggerAttributeAnalysis
```

**Notes**

If you get a successful response, then you triggered the analysis successfully. The backend analysis might run for some time. However, a successful trigger does not guarantee a successful analysis.

## Get variable analysis results

Gets a set of variable analysis results from previous analysis runs of a variable.

**URL**

/ibm/pq/service/getVariableAnalysisResult

**Method**

The request type POST

**URL parameters**

None

**Headers**

Content-Type: application/json

APIKEY: <encrypted key>. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

Username: <String>. Used to identify tenant and user. Mandatory.

variable_name: <String>. Contains the name of the variable to be analyzed.

dim_name: <String>. Contains the dimension names, which are separated by commas. Optional.

dim_value: <String>. Contains the dimension values, which are separated by commas. Optional.

runtime: <Runtime>. Timestamp. Optional. Default is the latest successfully analyzed time. Accepted format is yyyy-MM-dd HH:mm:ss.

**Sample body**

```
[{"Username":"<IBMid>","variable_name":"Height","dim_name":"DEVICE_ID",
"dim_value":"PMQ_PQ_DEVICE1","runtime":"2017-01-26 16:30:07"}]
```

**Success response**

201 Created

```
[{"dim_name":"DEVICE_ID","dim_value":"PMQ_PQ_DEVICE1",
"profile":{"threshold_upper":2.5,"variable_color":"3",
"lastbadptvalue":1177.3249999999998,"severity":728.9319462491873,
"num_points":200,"unaccept_level":5.55,"accept_level":5.0,"sigma":0.45,
"false_alarm_rate":5000.0,"target":5.0,"runtime":"2017-01-26 16:30:07",
"type_of_control":1.0,"forgiveness":0.0,"lastbadptindex":16,"rc":-2797.0,
"rcun":-1.0,"threshold_lower":null},"kpi":{"scheme1_val":[67.625,
135.14999999999998,204.77499999999998,274.4,345.025,416.65,486.275,
554.9,625.525,693.15,760.675,828.3,895.8249999999999,965.4499999999999,
1035.0749999999998,1105.6999999999998,1177.3249999999998],"scheme2_val":
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
"violation_type":["3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0",
"3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0"],"variable_val":[72.9,72.8,
74.9,74.9,75.9,76.9,74.9,73.9,75.9,72.9,72.8,72.9,72.8,74.9,74.9,75.9,76.9],
"variable_timestamp":["2017-01-23 13:14:35","2017-01-23 13:14:45",
"2017-01-23 13:14:55","2017-01-23 13:15:05","2017-01-23 13:15:15",
"2017-01-23 13:15:25","2017-01-23 13:15:35","2017-01-23 13:15:45",
"2017-01-23 13:15:55","2017-01-24 11:36:10","2017-01-24 11:36:20",
"2017-01-24 17:52:19","2017-01-24 17:52:29","2017-01-24 17:52:39",
"2017-01-24 17:52:49","2017-01-24 17:52:59","2017-01-24 17:53:09"]},
"history_result":[{"variable_color":"3","runtime":"2017-01-26 15:32:16"},
{"variable_color":"3","runtime":"2017-01-26 15:40:13"},
{"variable_color":"3","runtime":"2017-01-26 15:47:24"},
{"variable_color":"3","runtime":"2017-01-26 15:49:50"},
{"variable_color":"3","runtime":"2017-01-26 16:08:02"},
{"variable_color":"3","runtime":"2017-01-26 16:30:07"}]}]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "dim_name":"DEVICE_ID",
        "dim_value":"PMQ_PQ_DEVICE1",
        "error_message":"<error messages content. >"
    }
]
```

**Response items**

`dim_name`: <String>. Same as the input.

`dim_value`: <String>. Same as the input.

`profile`: <JSONObject>. Same as the `QEWSVariable` service response.

`kpi`: <JSONObject>. Same as the `QEWSVariable` service response.

`history_result`: <JSONArray>. Contains the history of successfully analyzed `runtime` and `variable_color`.

`error_message`: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>",
"variable_name":"Height","dim_name":"DEVICE_ID",
"dim_value":"PMQ_PQ_DEVICE1","runtime":"2017-01-26 16:30:07"}]'
-v "https://<serverhost>/ibm/pq/service/getVariableAnalysisResult"
```

## Get attribute analysis results

Gets a set of attribute analysis results from previous analysis runs of an attribute.

**URL**

/ibm/pq/service/getAttributeAnalysisResult

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`attribute_name: <String>`. Contains the name of the attribute to be analyzed.

`dim_name: <String>`. Contains the dimension names, which are separated by commas. Optional.

`dim_value: <String>`. Contains the dimension values, which are separated by commas. Optional.

`rundate: <Rundate>`. Rundate. Optional. Default is the latest successfully analyzed date. Accepted format is `yyyy-MM-dd`.

**Sample body**

```
[{"Username":"<IBMid>","attribute_name":"PRODUCT400",
"dim_name":"MODEL,PROCESS,SUPPLIER",
"dim_value":"BBB,PROCESS001,SUPA","rundate":"2016-11-12"}]
```

**Success response**

201 Created

```
[{"dim_name":"DEVICE_ID","dim_value":"PMQ_PQ_DEVICE1",
"profile":{"threshold_upper":2.5,"variable_color":"3",
"lastbadptvalue":1177.3249999999998,"severity":728.9319462491873,
"num_points":200,"unaccept_level":5.55,"accept_level":5.0,"sigma":0.45,
"false_alarm_rate":5000.0,"target":5.0,"runtime":"2017-01-26 16:30:07",
"type_of_control":1.0,"forgiveness":0.0,"lastbadptindex":16,"rc":-2797.0,
"rcun":-1.0,"threshold_lower":null},"kpi":{"scheme1_val":[67.625,
135.14999999999998,204.77499999999998,274.4,345.025,416.65,486.275,
554.9,625.525,693.15,760.675,828.3,895.8249999999999,965.4499999999999,
1035.0749999999998,1105.6999999999998,1177.3249999999998],"scheme2_val":
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
"violation_type":["3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0",
"3.0","3.0","3.0","3.0","3.0","3.0","3.0","3.0"],"variable_val":[72.9,72.8,
74.9,74.9,75.9,76.9,74.9,73.9,75.9,72.9,72.8,72.9,72.8,74.9,74.9,75.9,76.9],
"variable_timestamp":["2017-01-23 13:14:35","2017-01-23 13:14:45",
"2017-01-23 13:14:55","2017-01-23 13:15:05","2017-01-23 13:15:15",
"2017-01-23 13:15:25","2017-01-23 13:15:35","2017-01-23 13:15:45",
"2017-01-23 13:15:55","2017-01-24 11:36:10","2017-01-24 11:36:20",
"2017-01-24 17:52:19","2017-01-24 17:52:29","2017-01-24 17:52:39",
"2017-01-24 17:52:49","2017-01-24 17:52:59","2017-01-24 17:53:09"]},
"history_result":[{"variable_color":"3","runtime":"2017-01-26 15:32:16"},
{"variable_color":"3","runtime":"2017-01-26 15:40:13"},
{"variable_color":"3","runtime":"2017-01-26 15:47:24"},
{"variable_color":"3","runtime":"2017-01-26 15:49:50"},
{"variable_color":"3","runtime":"2017-01-26 16:08:02"},
{"variable_color":"3","runtime":"2017-01-26 16:30:07"}]}]
```

```
[{"dim_name":"MODEL,PROCESS,SUPPLIER","dim_value":"BBB,PROCESS001,SUPA",
"profile":{"sup_alarm1_flag":0,"severity_sup_alarm1":0.0,
"sup_alarm2_flag":0,"severity":0.9994,"numberofvintages":100,
"rundate":"2017-02-24","lam0":0.025,"prob":0.99,"lam1":0.0425,
"attribute_color":"4","forgiveness":7.0,"severity_sup_alarm2":0.0,
"last_bad_point":41,"rcun":-106.00007,"threshold":12.015587744868661},
"kpi":{"vintage_no":[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42],
"pct_failed_cumulate":[2.1052631578947367,4.912280701754385,
8.421052631578947,10.175438596491228,13.333333333333332,16.842105263157894,
20.0,23.859649122807017,26.31578947368421,28.421052631578945,
31.57894736842105,33.68421052631579,35.92982456140351,38.45614035087719,
40.14035087719298,42.94736842105263,44.070175438596486,45.19298245614035,
47.719298245614034,51.08771929824561,53.614035087719294,56.421052631578945,
58.10526315789473,60.91228070175438,62.59649122807017,65.68421052631578,
68.21052631578947,69.05263157894737,71.57894736842105,73.89473684210526,
75.78947368421052,78.73684210526315,80.21052631578947,81.68421052631578,
83.78947368421052,86.3157894736842,88.21052631578947,89.68421052631578,
90.3157894736842,93.26315789473684,97.68421052631578,100.0],
"failed_qty":[30.0,40.0,50.0,25.0,45.0,50.0,45.0,55.0,35.0,30.0,45.0,30.0,
32.0,36.0,24.0,40.0,16.0,16.0,36.0,48.0,36.0,40.0,24.0,40.0,24.0,44.0,36.0,
12.0,36.0,33.0,27.0,42.0,21.0,21.0,30.0,36.0,27.0,21.0,9.0,42.0,63.0,33.0],
"violation":["false","false","false","false","false","false","false",
"false","false","false","false","false","false","false","false","false",
"false","false","false","false","false","false","false","false","false",
"false","false","false","false","false","false","false","false","false",
"false","false","false","false","false","false","true","false"],
"failure_rate":[0.015915119363395226,0.031746031746031744,
0.025906735751295335,0.01557632398753894,0.03557312252964427,
0.02481389578163772,0.03571428571428571,0.027227722772277228,
0.01856763925729443,0.023076923076923078,0.02631578947368421,
0.02666666666666667,0.020779220779220778,0.030201342281879196,
0.026905829596412557,0.024875621890547265,0.018957345971563982,
0.017777777777777778,0.03180212014134275,0.0335195530726257,
0.026392961876832845,0.035211267605633804,0.027149321266968326,
0.026881720430107527,0.030927835051546393,0.029891304347826088,
0.026627218934911243,0.020202020202020204,0.030690537084398978,
0.031161473087818695,0.02631578947368421,0.04,0.038461538461538464,
0.03333333333333333,0.026455026455026454,0.0332409972299169,
0.025139664804469275,0.03271028037383177,0.024793388429752067,
0.03218390804597701,0.04375,0.026829268292682926],
"failure_rate_vs_produced":[0.013793103448275862,0.024390243902439025,
0.020242914979757085,0.013477088948787063,0.01859504132231405,
0.024630541871921183,0.021479713603818614,0.02576112412177986,
0.01707317073170732,0.015584415584415584,0.025069637883008356,
0.015873015873015872,0.02,0.02459016393442623,0.013605442176870748,
0.020833333333333332,0.010230179028132993,0.010126582278481013,
0.02727272727272727,0.03208556149732621,0.02349869451697128,
0.0228310502283105,0.014598540145985401,0.02386634844868735,
0.017804154302670624,0.022132796780684104,0.019027484143763214,
0.01084010840108401,0.025806451612903226,0.02546296296296296,
0.02097902097902098,0.04,0.038461538461538464,0.03333333333333333,
0.026455026455026454,0.0332409972299169,0.025139664804469275,
0.03271028037383177,0.024793388429752067,0.03218390804597701,0.04375,
0.026829268292682926],
"produced_qty":[2175.0,1640.0,2470.0,1855.0,2420.0,2030.0,2095.0,2135.0,
2050.0,1925.0,1795.0,1890.0,1600.0,1464.0,1764.0,1920.0,1564.0,1580.0,
1320.0,1496.0,1532.0,1752.0,1644.0,1676.0,1348.0,1988.0,1892.0,1107.0,
1395.0,1296.0,1287.0,1050.0,546.0,630.0,1134.0,1083.0,1074.0,642.0,363.0,
1305.0,1440.0,1230.0],
"vintage_date":["2016-10-01","2016-10-02","2016-10-03","2016-10-04",
"2016-10-05","2016-10-06","2016-10-07","2016-10-08","2016-10-09",
"2016-10-10","2016-10-11","2016-10-12","2016-10-13","2016-10-14",
"2016-10-15","2016-10-16","2016-10-17","2016-10-18","2016-10-19",
"2016-10-20","2016-10-21","2016-10-22","2016-10-23","2016-10-24",
"2016-10-25","2016-10-26","2016-10-27","2016-10-28","2016-10-29",
"2016-10-30","2016-10-31","2016-11-01","2016-11-02","2016-11-03",
"2016-11-04","2016-11-05","2016-11-06","2016-11-07","2016-11-08",
"2016-11-09","2016-11-10","2016-11-11"],
"cusum":[0.0,0.0,0.0,0.0,3.2805854085475525,0.0,3.4454842804505206,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.7729630869882127,0.0,
2.534976303644283,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,7.371236900375437,
10.364280088570665,10.587022228795925,3.187958081201395,3.4708624270172037,
0.0,0.0,0.0,0.0,15.50912489194345,7.944002403811818],
"pct_tested_cumulate":[3.6516146529513183,6.092481741926735,
9.831270219484319,12.940469963774433,15.391023033261657,19.29447317951996,
21.735340268495378,25.648476395265494,29.30009104821681,31.818445981286686,
35.13105131632475,37.31039693148137,40.293678929117995,42.60281668313283,
44.33079560643924,47.44580693903644,49.080800449429496,50.82427694155479,
53.017182929427946,55.79124774800953,58.433583231630536,60.63423800391313,
62.34671935840065,65.22926715871448,66.73253133414696,69.584083996823,
72.20317312721566,73.35386761201836,75.62619864008833,77.67768931248911,
79.66525251351194,81.69930842099146,82.7570174928808,83.97745103736851,
86.17423141744638,88.27221479630383,90.35276341024002,91.59644330795606,
92.29964549311326,94.82768640669495,97.61724879409542,100.0],
"pct_failed":[2.1052631578947367,2.807017543859649,3.508771929824561,
1.7543859649122806,3.1578947368421053,3.508771929824561,3.1578947368421053,
3.859649122807017,2.4561403508771926,2.1052631578947367,3.1578947368421053,
2.1052631578947367,2.245614035087719,2.526315789473684,1.6842105263157894,
2.807017543859649,1.1228070175438596,1.1228070175438596,2.526315789473684,
3.3684210526315788,2.526315789473684,2.807017543859649,1.6842105263157894,
2.807017543859649,1.6842105263157894,3.087719298245614,2.526315789473684,
0.8421052631578947,2.526315789473684,2.3157894736842106,1.894736842105263,
2.9473684210526314,1.4736842105263157,1.4736842105263157,2.1052631578947367,
2.526315789473684,1.894736842105263,1.4736842105263157,0.631578947368421,
2.9473684210526314,4.421052631578947,2.3157894736842106],
"tested_qty":[1885.0,1260.0,1930.0,1605.0,1265.0,2015.0,1260.0,2020.0,
1885.0,1300.0,1710.0,1125.0,1540.0,1192.0,892.0,1608.0,844.0,900.0,1132.0,
1432.0,1364.0,1136.0,884.0,1488.0,776.0,1472.0,1352.0,594.0,1173.0,1059.0,
1026.0,1050.0,546.0,630.0,1134.0,1083.0,1074.0,642.0,363.0,1305.0,1440.0,
1230.0]},
"history_result":[{"rundate":"2016-11-12","attribute_color":"2"},
{"rundate":"2017-02-23","attribute_color":"4"},
{"rundate":"2017-02-24","attribute_color":"4"}]}]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "dim_name":"DEVICE_ID",
        "dim_value":"PMQ_PQ_DEVICE1",
        "error_message":"<error messages content. >"
    }
]
```

**Response items**

`dim_name`: <String>. Same as the input.

`dim_value`: <String>. Same as the input.

`profile`: <JSONObject>. Same as the `QEWSAttribute` service response.

`kpi`: <JSONObject>. Same as the `QEWSAttribute` service response.

`history_result`: <JSONArray>. Contains the history of successfully analyzed `rundate` and `attribute_color`.

`error_message`: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>",
"attribute_name":"PRODUCT400","dim_name":"MODEL,PROCESS,SUPPLIER",
"dim_value":"BBB,PROCESS001,SUPA","rundate":"2016-11-12"}]'
-v "https://<serverhost>/ibm/pq/service/getAttributeAnalysisResult"
```

## Enable QEWS multivariate analysis on multiple variables independently

Enables the IBM Quality Early Warning System multivariate analysis for multiple variables' data independently from your other data on IBM Prescriptive Quality for Manufacturing Private. The event data itself and the parameters must be provided directly in the request. The analysis result is included in the response directly.

**URL**

/ibm/pq/service/QEWSMultivariate

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`VARIABLE_GROUP_NAME: <String>`. The name of the group of variables to be analyzed.

`EVENT_TIME: <JSONArray>`. Timestamp values in the sequence that you want to analyze data. Accepted format is `yyyy-MM-dd HH:mm:ss`.

`RUNTIME: <String>`. Runtime. Mandatory. The time point from when time is counted back for the number of points that are included in the analysis. Accepted format is `yyyy-MM-dd HH:mm:ss`.

`VARIABLES: <JSONArray>`. Event data of each variable in the group to be analyzed. Includes the following items:

- `VARIABLE_NAME: <String>`. The name of the variable.
- `VARIABLE_VALUE: <JSONArray>`. Numeric values in the same sequence of timestamps.
- `PARAMETER: <JSONObject>`. Parameters for each variable, including the following items:
  - `QEWS_TARGET: <Double>`. Mandatory.
  - `QEWS_SIGMA: <Double>`. Mandatory.
  - `QEWS_ACCEPT_LEVEL: <Double>`. Mandatory.
  - `QEWS_UNACCEPT_LEVEL: <Double>`. Mandatory.
  - `QEWS_TYPE_OF_CONTROL: <Integer>`. Optional. Default is 2.

- QEWS_FALSE_ALARM_RATE: <Double>. Optional. Default is 5000.

GROUP_PARAMETER: <JSONObject>. Contains parameters for doing the QEWS multivariate analysis, including the following items:

- QEWS_MEMBERS: <JSONArray>. Mandatory. Contains the variables in the group to be analyzed.
- QEWS_CORREL: <String>. Mandatory. List of partial rows in the correlation matrix R of the members. For every row, only elements after the diagonal element are included. For a p-dimensional group this list will contain (p-1) components, with the total number of p*(p-1)/2 elements. For a 1-dimensional group the list is empty, that is, it contains zero components. For example, If there are three variables in the group, and the correlation matrix R is:

  $$\mathbf{R} = \begin{pmatrix} 1 & 0.6 & 0.5 \\ 0.6 & 1 & -0.3 \\ 0.5 & -0.3 & 1 \end{pmatrix}$$

  then you should input the following string in the QEWS_CORREL item: `(0.6 0.5)(-0.3)`
- QEWS_TARGET: <Double>. Mandatory. The typical value of Mahalanobis distance λ between μ and $\mu_0$ under acceptable operating conditions. The formula of Mahalanobis distance λ is:

  $$\lambda = \sqrt{(\boldsymbol{\mu} - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)}$$

  where:
  - $\mu_0$ is the multivariate mean of the target population, that is, vector of targets for the individual variables.
  - μ is the multivariate mean of the actual population.
  - Σ is the covariance matrix that can be calculated from the standard deviations of the individual variables and their correlation matrix R.
- QEWS_ACCEPT_LEVEL: <Double>. Mandatory. The maximal level of λ that is still considered acceptable. This level is at or above the target, and it reflects the amount of allowable variation for λ above the target. In cases of low process capability, this level coincides with the target, which indicates the absence of any allowable variation for λ.
- QEWS_UNACCEPT_LEVEL: <Double>. Mandatory. The minimal level of λ that is still considered unacceptable. This is the level for which you want good detection capability. Generally, the unacceptable level should be further away from the target than the acceptable level. Keep a certain degree of separation between the acceptable and unacceptable levels. For a p-dimensional group, separation by √p is typically a good choice.
- QEWS_FALSE_ALARM_RATE: <Double>. Optional. Default is 5000, which means that the detection procedure produces a rate of false alarms of 1 per 5000 multivariate points when the distance λ between μ and $\mu_0$ is equal to QEWS_ACCEPT_LEVEL.
- QEWS_NUMBER_OF_POINTS: <Integer>. Optional. Default is 100.

**Sample body**

```
[{
 "Username": "<IBMid>",
 "VARIABLE_GROUP_NAME": "Anneal",
 "EVENT_TIME": ["2017-06-01 00:00:00", "2017-06-01 00:10:00"],
 "RUNTIME": "2017-06-02 00:00:00",
 "VARIABLES": [{
   "VARIABLE_NAME": "Ann_Str100",
   "VARIABLE_VALUE": ["0.1", "0.11"],
   "PARAMETER": {
    "QEWS_TARGET": "0.1",
    "QEWS_SIGMA": "0.02",
    "QEWS_ACCEPT_LEVEL": "0.1",
    "QEWS_UNACCEPT_LEVEL": "0.12",
    "QEWS_TYPE_OF_CONTROL": "1",
    "QEWS_FALSE_ALARM_RATE": "1000"
   }
  }, {
   "VARIABLE_NAME": "Ann_Time",
   "VARIABLE_VALUE": ["1.3", "1.4"],
   "PARAMETER": {
    "QEWS_TARGET": "1.3",
    "QEWS_SIGMA": "0.1",
    "QEWS_ACCEPT_LEVEL": "1.3",
    "QEWS_UNACCEPT_LEVEL": "1.4",
    "QEWS_TYPE_OF_CONTROL": "2",
    "QEWS_FALSE_ALARM_RATE": "1000"
   }
  },
  {
   "VARIABLE_NAME": "Ann_Temp",
   "VARIABLE_VALUE": ["2.8", "2.9"],
   "PARAMETER": {
    "QEWS_TARGET": "2.8",
    "QEWS_SIGMA": "0.2",
    "QEWS_ACCEPT_LEVEL": "2.8",
    "QEWS_UNACCEPT_LEVEL": "3",
    "QEWS_TYPE_OF_CONTROL": "2",
    "QEWS_FALSE_ALARM_RATE": "1000"
   }
  }
 ],
 "GROUP_PARAMETER": {
  "QEWS_MEMBERS": ["Ann_Str100", "Ann_Time", "Ann_Temp"],
  "QEWS_CORREL": "(0.6 0.5)(-0.3)",
  "QEWS_TARGET": "1",
  "QEWS_ACCEPT_LEVEL": "1",
  "QEWS_UNACCEPT_LEVEL": "2",
  "QEWS_FALSE_ALARM_RATE": "1000",
  "QEWS_NUMBER_OF_POINTS": "100"
 }
}]
```

**Success response**

```
201 Created
[
    {
        "result": {
            "kpi": {
                "scheme_val": [
                    0,
                    0.1511174574954226
                ],
                "violation_type": [
                    "0",
                    "0"
                ],
                "variable_val": [
                    0,
                    2.322893310794391
                ],
                "variable_timestamp": [
                    "2017-06-01 00:00:00",
                    "2017-06-01 00:10:00"
                ]
            },
            "profile": {
                "accept_level_input": "1",
                "accept_level": "1.849320433",
                "unaccept_level_input": "2",
                "false_alarm_rate": "1000",
                "lastbadptvalue": "0",
                "runtime": "2017-06-02 00:00:00",
                "severity": "0.29872858890998355",
                "forgiveness": "0",
                "lastbadptindex": "-1.0",
                "rc": "0",
                "rcun": "0",
                "threshold_lower": "NaN",
                "num_points": "100",
                "result_record": "2",
                "threshold_upper": "4.838980665718708",
                "sigma": "0.8090318373",
                "unaccept_level": "2.494231273",
                "variable_color": "1",
                "target": "1.849320433",
                "target_input": "1",
                "correl": "(0.6 0.5)(-0.3)",
                "unaccept_factor_sigma": "1.5",
                "members": [
                    "Ann_Str100",
                    "Ann_Time",
                    "Ann_Temp"
                ],
                "type_of_control": "1"
            }
        },
        "warning_message": ""
    }
]
```

If there is an error and a QEWS analysis cannot be done successfully, then the following response content is returned:

```
[
    {
        "error_message": "<error messages content. >"
    }
]
```

### Response items

`kpi`: `<JSONObject>`. Contains the first set of QEWS analysis results, which includes the violation type and evidence value of each analyzed event data point.

`violation_type`: `<JSONArray>`. Contains the violation type of each analyzed event data point.

- `0` indicates that there is no violation for the data point.
- `>0` indicates a violation.

`scheme_val`: `<JSONArray>`. The set of evidence values of each analyzed event data point.

variable_val: <JSONArray>. The Mahalanobis distance value of each analyzed event data point. This value can also be viewed as the biased estimate of Mahalanobis distance λ based exclusively on this event. It is calculated by the following formula.

$$\hat{\lambda}_i = \sqrt{(\mathbf{X}_i - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\mathbf{X}_i - \boldsymbol{\mu}_0)}, \quad i = 1,2,...$$

$\{X_i\}$ is the sequence of incoming multivariate observations. The new sequence $\{\hat{\lambda}_i\}$ reflects the underlying Mahalanobis distance λ. The statistics of $\{\hat{\lambda}_i\}$ form a sequence of biased estimates of λ.

variable_timestamp: <JSONArray>. The timestamps of each analyzed event data.

profile: <JSONObject>. The second set of QEWS analysis results, which includes the values of the QEWS parameters in the analysis, and the QEWS-calculated indicators, such as severity and forgiveness.

accept_level_input: <Double>. From the input parameter QEWS_ACCEPT_LEVEL.

accept_level: <Double>. The mean of the biased estimates of the Mahalanobis distance when the underlying true Mahalanobis distance is accept_level_input.

unaccept_level_input: <Double>. From the input parameter QEWS_UNACCEPT_LEVEL.

false_alarm_rate: <Integer>. From the input parameter QEWS_FALSE_ALARM_RATE. If the value does not exist in the input parameter, the default value is used.

lastbadptvalue: <Double>. The last bad point value. The Mahalanobis distance value of the last data point that is consistent with unacceptable behavior.

runtime: <String>. The value from the input parameter RUNTIME.

severity: <Double>. The degree of violation of acceptable conditions that are detected during analysis. This value is a base-10 logarithm of the "conformance to acceptable process level" test p-value.

forgiveness: <Integer>. The degree of forgiveness that is observed within the last acceptable period of the data. The value ranges from 0 to 9.

- 1 means that the degree of forgiveness is very low.
- 9 is achieved when the forgiveness criteria are satisfied.

lastbadptindex: <Integer>. The last bad point index. The index of the last data point that is consistent with unacceptable behavior.

rc: <Integer>. The return code.

- Zero means a normal run.
- Positive values correspond to error conditions.
- Negative values give the minutes that elapsed since the last point that was consistent with the unacceptable process level.

rcun: <Integer>. The second return code. The number of minutes that elapsed between the timestamp of the last bad point and the timestamp of the last data point.

threshold_lower: <NaN>. The QEWS calculated threshold for the lower direction evidence values. Since the multivariate analysis is only on the upper direction, the value is NaN, which means "not a number."

num_points: <Integer>. From the input parameter QEWS_NUMBER_OF_POINTS. If the value does not exist in the input parameter, the default value is used.

result_record: <Integer>. The number of data points that are analyzed.

threshold_upper: <Double>. The QEWS-calculated threshold for the evidence values.

sigma: <Double>. The standard deviation of the biased estimates of the Mahalanobis distance of the analyzed event data point.

unaccept_level: <Double>. The mean of the biased estimates of the Mahalanobis distance when the underlying true Mahalanobis distance is unaccept_level_input.

variable_color: <Integer>. Each number presents a status and color that is classified by QEWS.

- 1 means "Acceptable" and "Grey".
- 2 means "Caution" and "Amber" (Yellow).
- 3 means "Alert" and "Red".
- 4 means "Alert but improving" and "Blue".

target: <Double>. The mean of the biased estimates of the Mahalanobis distance when the underlying true Mahalanobis distance is target_input.

target_input: <Double>. From the input parameter QEWS_TARGET.

correl: <Double>. From the input parameter QEWS_CORREL.

unaccept_factor_sigma: <Double>. An input parameter of QEWS. The default value is used in all cases.

members: <JSONArray>. From the input parameter QEWS_MEMBERS.

type_of_control: <Integer>. The default value 1 is used for QEWS multivariate analysis.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

### Sample call

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;
charset=UTF-8" -k --data '[{"Username":"<IBMid>","VARIABLE_GROUP_NAME":
"Anneal","EVENT_TIME":["2017-06-01 00:00:00","2017-06-01 00:10:00"],
"RUNTIME":"2017-06-02 00:00:00","VARIABLES":[{"VARIABLE_NAME":
"Ann_Str100","VARIABLE_VALUE":["0.1","0.11"],"PARAMETER":
{"QEWS_TARGET":"0.1","QEWS_SIGMA":"0.02","QEWS_ACCEPT_LEVEL":
"0.1","QEWS_UNACCEPT_LEVEL":"0.12","QEWS_TYPE_OF_CONTROL":"1",
"QEWS_FALSE_ALARM_RATE":"1000"}},{"VARIABLE_NAME":"Ann_Time",
"VARIABLE_VALUE":["1.3","1.4"],"PARAMETER":{"QEWS_TARGET":"1.3",
"QEWS_SIGMA":"0.1","QEWS_ACCEPT_LEVEL":"1.3","QEWS_UNACCEPT_LEVEL":
"1.4","QEWS_TYPE_OF_CONTROL":"2","QEWS_FALSE_ALARM_RATE":"1000"}},
{"VARIABLE_NAME":"Ann_Temp","VARIABLE_VALUE":["2.8","2.9"],
"PARAMETER":{"QEWS_TARGET":"2.8","QEWS_SIGMA":"0.2","QEWS_ACCEPT_LEVEL":
"2.8","QEWS_UNACCEPT_LEVEL":"3","QEWS_TYPE_OF_CONTROL":"2",
"QEWS_FALSE_ALARM_RATE":"1000"}}],"GROUP_PARAMETER":{"QEWS_MEMBERS":
["Ann_Str100","Ann_Time","Ann_Temp"],"QEWS_CORREL":"(0.6 0.5)(-0.3)",
"QEWS_TARGET":"1","QEWS_ACCEPT_LEVEL":"1","QEWS_UNACCEPT_LEVEL":"2",
"QEWS_FALSE_ALARM_RATE":"1000","QEWS_NUMBER_OF_POINTS":"100"}}]'
-v "https://<serverhost>/ibm/pq/service/QEWSMultivariate"
```

**Notes**

This action is designed for normal-sized data processing. Do not exceed 5000 event data records.

The timestamps of the event data must be sorted by the right time sequence. The service does not sort the data by timestamps. If two events have the same timestamp, the service still handles them as two data points and analyzes them with the sequence that is provided in your data.

## Enable target setting for QEWS variable analysis on a variable independently

Enables you to perform the target setting procedures for IBM Quality Early Warning System variable analysis for a variable independently from your other data on IBM Prescriptive Quality for Manufacturing Private. All input data must be provided directly in the request. The suggested targets result is included in the response directly.

The purpose of the target setting procedures is to assist in setting QEWS variable analysis parameters, including QEWS_TARGET, QEWS_SIGMA, QEWS_ACCEPT_LEVEL, and QEWS_UNACCEPT_LEVEL. In many situations, these parameters are dictated by business requirements. However, if you want to use a historical capability-based target setting approach to assist in determining the parameter values, this API provides suggested values.

This API has three modes:

- Mode 0: You provide a set of event data in the request, and the API provides setting suggestions for QEWS_TARGET, QEWS_SIGMA, QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL in response.
- Mode 1: You provide QEWS_TARGET and a set of event data in the request, and the API provides setting suggestions for QEWS_SIGMA, QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL in response.
- Mode 2: You provide QEWS_TARGET and QEWS_SIGMA in the request, and the API provides setting suggestion for QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL in response.

In all the three modes, you must provide the following inputs:

- QEWS_TYPE_OF_CONTROL: This value is the same as the QEWS variables analysis input parameter.
- QEWS_SPEC_DEVIATION: A parameter used for this target setting procedure. It is defined as half of the distance between the upper and lower specification limits (specs). Sometimes it is based on engineering specifications; in other cases, it could stem from requirements on process capability indices, such as Cp or Cpk that are routinely produced within the framework of quality management and improvement programs. In the case of one-sided type of control, QEWS_SPEC_DEVIATION > 0 when we want to detect a change up, and QEWS_SPEC_DEVIATION < 0 when we want to detect a change down. In the case of two-sided detection, you can use any sign.
- QEWS_LIMITATION: A parameter used for this target setting procedure. It is used in the formula to set QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL.

$$\Delta_{acc} = \begin{cases} 0 \text{ if } d \leq a \\ g\sigma(d-a) \text{ if } d > a \end{cases}$$

$$\Delta_{unacc} = \begin{cases} \sigma d^2/a^2 \text{ if } d \leq a \\ \sigma(d-a+1) \text{ if } d > a \end{cases}$$

$\Delta_{acc}$ and $\Delta_{unacc}$ are the acceptable and unacceptable deviations of the process mean from the QEWS_TARGET.

$g$ is a constant. The default value of $g$ is 0.5.

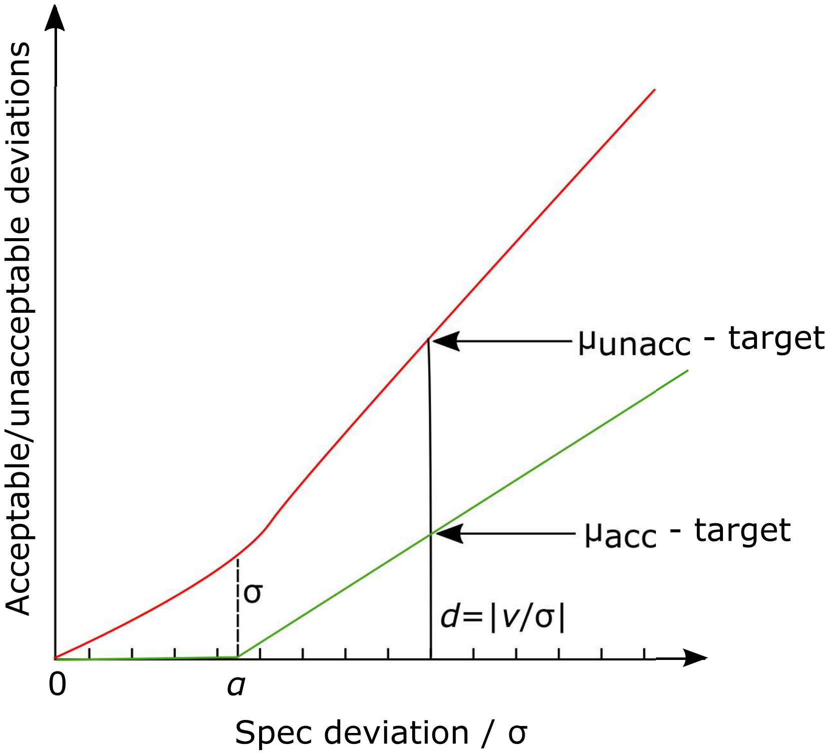
σ is QEWS_SIGMA.

$d = |v / \sigma|$, is the relative spec deviation. ($v$ is QEWS_SPEC_DEVIATION)

$a$ is QEWS_LIMITATION

The values $\Delta_{acc}$ and $\Delta_{unacc}$ are used with the plus sign (+) if $v$ >0 and the minus sign (-) if $v$ <0

The formula can be illustrated by the following chart.



**URL**

/ibm/pq/service/QEWSVariableTargetSetting

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`VARIABLE_NAME: <String>`. Contains the name of the variable to be set as the target.

`VARIABLE_VALUE: <JSONArray>`. Contains a set of numeric values of the event data for the variable. You must provide this information when you use mode 2 or mode 3. You must provide at least 3 data points in the array.

`PARAMETER: <JSONObject>`. Contains parameters for target setting procedures, including the following items:

- `QEWS_MODE: <Integer>.` Mandatory. The index of the mode of the target setting procedures. Valid values are 0, 1, and 2.
- `QEWS_TYPE_OF_CONTROL: <Integer>`. Optional. The default value is 2.
- `QEWS_SPEC_DEVIATION: <Double>`. Optional. The default value is 6 * QEWS_SIGMA. This is an important input parameter. Do not use the default setting unless you have no idea about the value. If you choose to use the default value, please study the variable to establish its contribution to quality characteristics and establishing specs.
- `QEWS_TARGET: <Double>`. You must provide this value when using mode 1 and mode 2.
- `QEWS_SIGMA: <Double>`. You must provide this value when using mode 2.
- `QEWS_LIMITATION: <Double>`. Optional. The default value is 5. This value must be positive.

**Sample body**

```
[{"Username":"<IBMid>","VARIABLE_NAME":"Offset","VARIABLE_VALUE":["3.46",
"3.64","3.47","3.69","3.27","2.20","2.95","3.11","3.22","3.11","2.95","3.35",
"3.59","3.47","3.51","3.43","3.66","3.53","2.96","2.99"],"PARAMETER":
{"QEWS_MODE":"1","QEWS_TYPE_OF_CONTROL":"1","QEWS_SPEC_DEVIATION":
"10","QEWS_TARGET":"2.8","QEWS_SIGMA":"0.35","QEWS_LIMITATION":"5"}}]
```

**Success response**

```
201 Created
[
    {
        "result": {
            "profile": {
                "variable_name": "Offset",
                "unaccept_level": "11.4088",
                "accept_level": "6.9305",
                "sigma": "0.34779999999999983",
                "target": "2.8",
                "limitation": "5.0",
                "spec_deviation": "10.0",
                "type_of_control": "1",
                "mode": "1"
            }
        },
        "warning_message": ""
    }
]
```

### Response items

profile: <JSONObject>. Contains the results for the target setting procedure of the QEWS variable.

variable_name: <String>. From the input parameter VARIABLE_NAME.

unaccept_level: <Double>. The suggested unacceptable level value from the target setting procedure.

accept_level: <Double>. The suggested acceptable level value from the target setting procedure.

sigma: <Double>. From the input parameter QEWS_SIGMA for mode 2 or the suggested value from the target setting procedure for mode 0 and mode 1.

target: <Double>. From the input parameter QEWS_TARGET for mode 1 and 2 or the suggested value from the target setting procedure for mode 0.

limitation: <Double>. From the input parameter QEWS_LIMITATION. If the value does not exist in the input parameter, the default value is used.

spec_deviation: <Double>. From the input parameter QEWS_SPEC_DEVIATION. If the value does not exist in the input parameter, the default value is used.

type_of_control: <Integer>. From the input parameter QEWS_TYPE_OF_CONTROL. If the value does not exist in the input parameter, the default value is used.

mode: <String>. From the input parameter QEWS_MODE.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

### Sample call

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;charset=UTF-8"
-k --data '[{"Username":"<IBMid>","VARIABLE_NAME":"Offset","VARIABLE_VALUE":
["3.46","3.64","3.47","3.69","3.27","2.20","2.95","3.11","3.22","3.11","2.95",
"3.35","3.59","3.47","3.51","3.43","3.66","3.53","2.96","2.99"],"PARAMETER":
{"QEWS_MODE":"1","QEWS_TYPE_OF_CONTROL":"1","QEWS_SPEC_DEVIATION":"10",
"QEWS_TARGET":"2.8","QEWS_SIGMA":"0.35","QEWS_LIMITATION":"5"}}]'
-v "https://<serverhost>/ibm/pq/service/QEWSVariableTargetSetting"
```

### Notes

This action is designed for normal-sized data processing. Do not exceed 5000 event data records.

## Trigger target setting for QEWS attribute analysis

Triggers the target setting procedure for IBM Quality Early Warning System (QEWS) analysis for attributes. The analysis uses event data and parameter data from IBM Prescriptive Quality for Manufacturing Private.

The purpose of the target setting procedure is to assist you in setting QEWS attribute analysis parameters, including QEWS_LAM0 and QEWS_LAM1. In many cases, these parameters are dictated by business requirements. However, you might prefer to use a historical capability-based target setting approach to aid in determining the parameter values. You can use this API to trigger the procedure of calculating suggestions or providing suggestions directly.

### URL

```
/ibm/pq/service/attributeTargetSetting
```

**Method**

The request type POST

**URL parameters**

**Username**: *String*. Used to identify the user.

**alpha**: *Double*. Trimming proportion alpha (typically, alpha = 0.1). This parameter controls the estimation of the process capability. For every given part (say, part A) the objective is to establish the failure rate of the part that the process can deliver. The data for part A consists of a sequence of failure rates and sample sizes, aggregated by date. It is necessary to eliminate extreme cases from the data, trimming away vintages (dates) for which the failure rates are either extremely good or extremely bad. The remaining data is then used to deliver the robust estimate of the underlying failure rate p for part A.

**beta**: *Double*. Confidence level $\beta$ (typically, $\beta = 0.9$) of the two-sided confidence bounds for underlying failure rate p of part A.

**minyards**: *Integer*. Minimal sample size needed to establish a yardstick. This value is referred to as minyards. Typically, minyards equals100. Yardstick computation requires us to borrow data from parts in higher levels of hierarchy if the sample sizes in lower levels of hierarchy are too small. So, minyards = 100 tells us that we should not use data from higher levels of hierarchy if we managed to accumulate a sample of size 100 using lower levels. We assume the order of dimensions in upload file as the level of hierarchy. For example, the level of our sample data hierarchy from high to low is MODEL>PROCESS>SUPPLIER.

**Headers**

```
Content-Type: application/json
```

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

None.

**Sample body**

The binary content of the data file. Only the attribute event data is supported. The DATE column must use one of the following date formats:

- yyyy/MM/dd
- yyyy/M/d
- dd.MM.yyyy
- MM-dd-yyyy
- M/d/yyyy
- yyyy-MM-dd

**Success response**

201 Created

```
{
 "error_message": "",
 "attributes": [{
  "QEWS_LAM0": 0.028143,
  "QEWS_LAM1": 0.04925025,
   "DIM_VALUE": "BBB,PROCESS001,SUPA",
   "QEWS_ATTRIBUTE_NAME": "PRODUCT400"
  }, {
   "QEWS_LAM0": 0.033455,
   "QEWS_LAM1": 0.058546249999999994,
   "DIM_VALUE": "AAA,PROCESS001,SUPA",
   "QEWS_ATTRIBUTE_NAME": "PRODUCT400"
  }, {
   "QEWS_LAM0": 0.031704,
   "QEWS_LAM1": 0.055482000000000004,
   "DIM_VALUE": "DDD,PROCESS001,SUPA",
   "QEWS_ATTRIBUTE_NAME": "PRODUCT500"
  }, {
   "QEWS_LAM0": 0.027483,
   "QEWS_LAM1": 0.04809525,
   "DIM_VALUE": "CCC,PROCESS001,SUPA",
   "QEWS_ATTRIBUTE_NAME": "PRODUCT500"
  }]
}
```

**Response items**

QEWS_ATTRIBUTE_NAME: <String>. From the ATTRIBUTE_NAME column in the input file.

DIM_VALUE: <String>. From the dimension columns in the input file.

QEWS_LAM0: <String>. The acceptable failure rate.

QEWS_LAM1: <String>. The unacceptable failure rate.

error_message: <String>. Error messages.

**Sample call**

```
curl -k -X POST -H "APIKEY:<encrypted key>" -H "Accept-Language:en-US,en;q=0.5"
-H "Content-Type:multipart/form-data" -H "Connection:keep-alive" -F
"data=@/tmp/Event_PRODUCT400500.csv" "https://<serverhost>/ibm/pq/service/
attributeTargetSetting?Username=<IBMid>&alpha=0.1&beta=0.9&minyards=100"
```

## Trigger target setting for QEWS variable analysis

Triggers the target setting procedure for IBM Quality Early Warning System (QEWS) analysis for variables. The analysis uses event data and parameter data from IBM Prescriptive Quality for Manufacturing Private.

The purpose of the target setting procedure is to assist you in setting QEWS variable analysis parameters, including QEWS_TARGET, QEWS_SIGMA, QEWS_ACCEPT_LEVEL, and QEWS_UNACCEPT_LEVEL. In many cases, these parameters are dictated by business requirements. However, you might prefer to use a historical capability-based target setting approach to aid in determining the parameter values. You can use this API to trigger the procedure of calculating suggestions or providing suggestions directly.

This API provides three modes:

- Mode 0: The API triggers the procedure of recommending the QEWS_TARGET, QEWS_SIGMA, QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL parameters by using event data of the variables in IBM Prescriptive Quality for Manufacturing Private. You can get the recommended parameter values by using the "Get target setting for QEWS variable analysis" API call.
- Mode 1: The API triggers the procedure of recommending the QEWS_SIGMA, QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL parameters by using event data of the variables in IBM Prescriptive Quality for Manufacturing

Private and the QEWS_TARGET value that is provided in the API request. You can get the recommended parameter values by using the "Get target setting for QEWS variable analysis" API call.

- Mode 2: The API runs the procedure of recommending QEWS_ACCEPT_LEVEL and QEWS_UNACCEPT_LEVEL by using the QEWS_TARGET and QEWS_SIGMA values that are provided in the API request. You can get the recommended parameter values directly in the response of this API call.

In all three modes, you must provide the following inputs:

- QEWS_TYPE_OF_CONTROL
- QEWS_SPEC_DEVIATION
- QEWS_LIMITATION

**URL**

```
/ibm/pq/service/triggerVariableTargetSetting
```

**Method**

The request type POST

**URL parameters**

None

**Headers**

`Content-Type: application/json`

`APIKEY: <encrypted key>`. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

`Username: <String>`. Used to identify tenant and user. Mandatory.

`variables: <JSONArray>`. Contains all of the variables for performing the target setting procedure for QEWS variable analysis, including the following variables:

- `variable_name: <String>`. Contains the name of the variable to be set as the target.
- `dim_name: <String>`. Contains the dimension names, which are separated by commas. The combination of dimension names is used to select the data for the procedure. Optional.
- `dim_value: <String>`. Contains the dimension values, which are separated by commas. The combination of dimension values is used to select the data for the procedure. Use the same sequence as `dim_name`. Optional.
- `runtime: <String>`. Timestamp. Mandatory. The time point from which time is counted back for the number of points that are included in the procedure. The accepted format is `yyyy-MM-dd HH:mm:ss`.

`PARAMETER: <JSONObject>`. Contains parameters for target setting procedures, including the following parameters:

- `QEWS_MODE: <Integer>`. Mandatory. The index of the mode of the target setting procedures. Only 0, 1, and 2 are accepted.
- `QEWS_TYPE_OF_CONTROL: <Integer>`. Optional. The default value is 2.
- `QEWS_SPEC_DEVIATION: <Double>`. Optional. The default value is 6 * QEWS_SIGMA.

- QEWS_TARGET: <Double>. This parameter must be provided when you use mode 1 or mode 2.
- QEWS_SIGMA: <Double>. This parameter must be provided when you use mode 2.
- QEWS_LIMITATION: <Double>. Optional. The default value is 5. This value must be positive.
- QEWS_NUMBER_OF_POINT: <Integer>. Optional. The maximum number of points in the procedure. This parameter is used only for mode 0 or 1. The default value is 10000.

**Sample body**

```
[{"Username":"<IBMid>","variables":[{"variable_name":"Offset","dim_name":
"Plant,Product","dim_value":"SHENZHEN,Product1","runtime":"2017-08-01 00:00:00",
"PARAMETER":{"QEWS_MODE":"2","QEWS_TYPE_OF_CONTROL":"2","QEWS_SPEC_DEVIATION":
"10","QEWS_TARGET":"2.8","QEWS_SIGMA":"0.35","QEWS_LIMITATION":"5"}},
{"variable_name":"Offset","dim_name":"Plant,Product","dim_value":"SHENZHEN,Product2",
"runtime":"2017-08-11 00:00:00","PARAMETER":{"QEWS_MODE":"0","QEWS_TYPE_OF_CONTROL":
"2","QEWS_SPEC_DEVIATION":"10","QEWS_LIMITATION":"5","QEWS_NUMBER_OF_POINTS":"10000"}},
{"variable_name":"Offset","dim_name":"Product","dim_value":"Product1","runtime":
"2017-08-21 00:00:00","PARAMETER":{"QEWS_MODE":"1","QEWS_TYPE_OF_CONTROL":"2",
"QEWS_SPEC_DEVIATION":"10","QEWS_TARGET":"2.8","QEWS_LIMITATION":"5"}}]}]
```

**Success response**

201 Created

```
[
    {
        "variables": [
            {
                "result": {
                    "profile": {
                        "variable_name": "Offset",
                        "unaccept_level": "11.399999999999999",
                        "accept_level": "6.925",
                        "sigma": "0.35",
                        "target": "2.8",
                        "limitation": "5.0",
                        "spec_deviation": "10.0",
                        "type_of_control": "2",
                        "mode": "2"
                    }
                },
                "variable_name": "Offset",
                "runtime": "2017-08-01 00:00:00",
                "dim_value": "SHENZHEN,Product1",
                "dim_name": "Plant,Product"
            },
            {
                "result": "triggered",
                "variable_name": "Offset",
                "runtime": "2017-08-11 00:00:00",
                "dim_value": "SHENZHEN,Product2",
                "dim_name": "Plant,Product"
            },
            {
                "result": "triggered",
                "warning_message": "QEWS parameter 'QEWS_NUMBER_OF_POINTS' is missing. The
                "variable_name": "Offset",
                "runtime": "2017-08-21 00:00:00",
                "dim_value": "Product1",
                "dim_name": "Product"
            }
        ]
    }
]
```

**Response items**

variable_name: <String>. From the input parameter variable_name.

dim_name: <String>. From the input parameter dim_name.

dim_value: <String>. From the input parameter dim_value.

runtime: <String>. From the input parameter runtime.

result: <String>. Contains the result of the QEWS variable target setting procedure for mode 2, or the response about whether the procedure is triggered for mode 0 or 1.

profile: <JSONObject>. Contains the results for the QEWS variable target setting procedure. This value is the same as the QEWSVariableTargetSetting service response. This response item is provided directly only when mode 2 is used.

warning_message: <String>. Warning messages.

error_message: <String>. Error messages.

**Sample call**

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;charset=UTF-8"
-k --data '[{"Username":"<IBMid>","variables":[{"variable_name":"Offset","dim_name":
"Plant,Product","dim_value":"SHENZHEN,Product1","runtime":"2017-08-01 00:00:00",
"PARAMETER":{"QEWS_MODE":"2","QEWS_TYPE_OF_CONTROL":"2","QEWS_SPEC_DEVIATION":
"10","QEWS_TARGET":"2.8","QEWS_SIGMA":"0.35","QEWS_LIMITATION":"5"}},{"variable_name":
"Offset","dim_name":"Plant,Product","dim_value":"SHENZHEN,Product2","runtime":
"2017-08-11 00:00:00","PARAMETER":{"QEWS_MODE":"0","QEWS_TYPE_OF_CONTROL":"2",
"QEWS_SPEC_DEVIATION":"10","QEWS_LIMITATION":"5","QEWS_NUMBER_OF_POINTS":"10000"}},
{"variable_name":"Offset","dim_name":"Product","dim_value":"Product1","runtime":
"2017-08-21 00:00:00","PARAMETER":{"QEWS_MODE":"1","QEWS_TYPE_OF_CONTROL":
"2","QEWS_SPEC_DEVIATION":"10","QEWS_TARGET":
"2.8","QEWS_LIMITATION":"5"}}]}]'
-v "https://<serverhost>/ibm/pq/service/triggerVariableTargetSetting"
```

## Get target setting for QEWS variable analysis

Gets a set of variable target setting results from previous target setting runs.

**URL**

/ibm/pq/service/getVariableTargetSetting

**Method**

The request type POST

**URL parameters**

None

**Headers**

Content-Type: application/json

APIKEY: <encrypted key>. The API key.

**Note:** The system uses the APIKEY to check with the provisioning console to do authentication. You must provide an APIKEY in the request header.

**Data parameters**

Username: <String>. Used to identify tenant and user. Mandatory.

variable_name: <String>. Contains the name of the variable to be set as the target.

`dim_name`: <String>. Contains the dimension names, which are separated by commas. Optional.

`dim_value`: <String>. Contains the dimension values, which are separated by commas. Use the same sequence as `dim_name`. Optional.

`runtime`: <String>. Timestamp. Optional. The default value is the latest successfully analyzed time. The accepted format is `yyyy-MM-dd HH:mm:ss`.

**Sample body**

```
[{"Username":"<IBMid>","variable_name":"Offset","dim_name":"Plant,Product",
"dim_value":"SHENZHEN,Product1","runtime":"2017-08-01 00:00:00"}]
```

**Success response**

201 Created

```
[
    {
        "variable_name": "Offset",
        "dim_name": "Plant,Product",
        "dim_value": "SHENZHEN,Product1",
        "history_result": [
            {
                "status": "succeeded",
                "runtime": "2017-08-01 00:00:00"
            },
            {
                "status": "succeeded",
                "runtime": "2017-08-20 00:00:00"
            }
        ],
        "runtime": "2017-08-01 00:00:00",
        "status": "succeeded",
        "profile": {
            "sigma": "0.35",
            "unaccept_level": "11.399999999999999",
            "accept_level": "6.925",
            "target": "2.8",
            "limitation": "5.0",
            "spec_deviation": "10.0",
            "type_of_control": "2",
            "mode": "2"
        }
    }
]
```

**Response items**

`variable_name`: <String>. From the input parameter `variable_name`.

`dim_name`: <String>. From the input parameter `dim_name`.

`dim_value`: <String>. From the input parameter `dim_value`.

`runtime`: <String>. From the input parameter `runtime`. If the value does not exist in the input parameter, the value of the latest runtime is used.

`history_result`: <String>. Contains the list of history results of the QEWS variable target setting procedure for the variable and dimensions. Each history result includes status and runtime information.

`profile`: <JSONObject>. Contains the results of the QEWS variable target setting procedure. This value is the same as the `QEWSVariableTargetSetting` service response.

`status`: <String>. The status of the triggered target setting procedure at each runtime.

`warning_message`: `<String>`. Warning messages.

`error_message`: `<String>`. Error messages.

### Sample call

```
curl -H "APIKEY:<encrypted key>" -H "Content-Type:application/json;charset=UTF-8"
-k --data '[{"Username":"<IBMid>","variable_name":"Offset","dim_name":"Plant,Product",
"dim_value":"SHENZHEN,Product1","runtime":"2017-08-01 00:00:00"}]'
-v "https://<serverhost>/ibm/pq/service/getVariableTargetSetting"
```

# Chapter 7. Troubleshooting

This section contains troubleshooting information for Prescriptive Quality for Manufacturing Private

## Cannot open the Prescriptive Quality for Manufacturing Private web site

The URL that was provided by your administrator does not open.

In your browser, add the URL to the list of trusted sites, clear the browsing history cache, and try again.

## File upload or parsing fails

When you attempt to upload an event file or a parameter file, the upload or parse fails.

Event and parameter files must be formatted correctly.

If the upload failed, correct the problems in the file, save the file, and upload it again. If parsing failed, decide whether you want to keep the file on the system; if not, delete it and upload the correct file.

**Related tasks**:

"Uploading data files and previewing the data" on page 16
On the home page, you can upload files for variable analysis and attribute analysis. Variable analysis and attribute analysis each require an event data file and an IBM Quality Early Warning System parameter file.

## Event file is identified as a variable parameter file

An event file is mistakenly identified as a variable parameter file.

The event file must not contain any of the following column names (case-insensitive), which are reserved exclusively for variable parameter files:

- QEWS_VARIABLE or QEWS_VARIABLE_NAME
- QEWS_TARGET
- QEWS_ACCEPT_LEVEL
- QEWS_UNACCEPT_LEVEL
- QEWS_SIGMA

After the column names are removed, save the file and upload it again. It will be correctly identified as an event file.

## Event file is identified as an attribute parameter file

An event file is mistakenly identified as an attribute parameter file.

The event file must not contain any of the following column names (case-insensitive), which are reserved exclusively for attribute parameter files:

- QEWS_ATTRIBUTE_NAME
- QEWS_LAM0
- QEWS_LAM1

After the column names are removed, save the file and upload it again. It will be correctly identified as an event file.

## Variable event file is identified as an attribute event file

A variable event file is mistakenly identified as an attribute event file.

The variable event file must not contain any of the following column names (case-sensitive), which are reserved exclusively for attribute event files:

- ATTRIBUTE_NAME
- DATE
- PRODUCED_QTY
- TESTED_QTY
- FAILED_QTY

After the column names are removed, save the file and upload it again. It will be correctly identified as a variable event file.

## Parameters are missing for a variable

Parameters are missing for a variable.

Confirm that you uploaded a parameter file that contains the parameters that are required for the variable. The variable name must be in the *QEWS_VARIABLE* or *QEWS_VARIABLE_NAME* column. You can upload a new parameter file that contains the required values, or you can edit the parameters in the user interface by clicking the variable.

## Parameters are missing for an attribute

Parameters are missing for an attribute.

Confirm that you uploaded an attribute parameter file that contains the parameters that are required for the attribute. The attribute name must be in the *QEWS_ATTRIBUTE_NAME* column. You can upload a new parameter file that contains the required values, or you can edit the parameters in Prescriptive Quality for Manufacturing Private by clicking the attribute.

## Variable analysis fails

In Notifications, the message appears: `An error occurred when analyzing the variable`.

The error might have one of the following causes:

- The **`target`**, **`sigma`**, **`accept_level`**, **`unaccept_level`**, **`type_of_control`**, **`false_alarm_rate`**, or **`number_of_points`** parameter values are not numbers. Change the values to numbers.

- The **sigma** parameter value is less than or equal to 0. Change the value to be greater than 0.
- The **target**, **accept_level**, or **unaccept_level** parameter values do not satisfy the following requirements: If **accept_level** is greater than **target**, then **unaccept_level** must be greater than or equal to **accept_level**. If **accept_level** is less than **target**, then **unaccept_level** must be less than **accept_level**.
- The **false_alarm_rate** parameter value is less than 1. Change the value to be greater than or equal to 1.
- The **type_of_control** parameter value is not in the range (1,2). Change the value to be in the range (1,2).
- The **number_of_points** parameter value is less than 1. Change the value to be greater than or equal to 1.
- The **runtime** parameter value uses an unsupported date format. It must use the yyyy-MM-dd HH:mm:ss date format.
- The variable event data contains too many data points or dimension values. Reduce the number of variables or reduce the number of data points in each variable.
- Apache HBase is not running correctly, Apache Kafka is not running correctly, or tenant permissions are not set correctly. Contact your administrator to correct these issues.

## Attribute analysis fails

In Notifications, the message appears: `An error occurred when analyzing the attribute`.

The error might have one of the following causes:

- The **lam0**, **lam1**, **prob**, or **number_of_vintages** parameter values are not numbers. Change the values to numbers.
- The **lam0** parameter value is greater than or equal to **lam1**. Change the value to be less than **lam1**.
- The **prob** parameter value is not in the range (0,1). Change the value to be in the range (0,1).
- The **number_of_ vintages** parameter value is less than 1. Change the value to be greater than or equal to 1.
- The **rundate** parameter value uses an unsupported date format. It must use the yyyy-MM-dd date format.
- The attribute event data contains too many data points or dimension values. Reduce the number of attributes or reduce the number of data points in each attribute.
- Apache HBase is not running correctly, Apache Kafka is not running correctly, or tenant permissions are not set correctly. Contact your administrator to correct these issues.

## Report page is blank

When you view analysis reports, the report page is blank.

Nothing is shown in the report page if no successful run exists for any date for object and dimension values.

## Cannot create or connect an IoT integration

When you attempt to create or connect an IoT integration, the operation fails.

Connect to Watson IoT Platform by using the same organization ID. Verify that the API key and authentication token values that are used to connect to the platform are correct.

## Cannot receive Watson IoT Platform data or parsing fails

Data from Watson IoT Platform is not received, for example, no event file is created, or parsing of an IoT event file fails.

If Watson IoT Platform data is not received, complete the following steps:

- Verify that the integration is in connected status.
- Verify that the correct devices are selected in the related integration.
- Delete and then re-create the integration.

If the IoT event file fails to parse, complete the following steps:

- Check the event file to ensure that data is formatted properly.
- If devices are sending improperly formatted data, correct the issue or revise the data in the CSV file and reupload the file.

## Analysis does not include data collected from Watson IoT Platform

Data is present in the event files, but analysis does not include the data. Or, analysis is configured to run periodically, but does not run according to the analysis interval setting.

If data is present in the event files, but analysis does not include the data, complete the following steps to troubleshoot this issue:

- Verify that the IoT event files include the correct object and the required dimensions of the analysis.
- Verify that the analysis is scheduled to run after the time stamps in the IoT event files. For example, the analysis might be configured to run in the past. Or, the time stamp in the event file, which is the time on the server where the product is hosted, might be different from the local time.

If analysis is configured to run periodically, but does not run according to the analysis interval setting, check your notifications for more information. The analysis does not run in the following situations:

- For variable analysis, there is no new data and no changes to the parameters. For attribute analysis, there is no new data, no changes to the parameters, and no change to the analysis interval setting.
- There are no event files for the analysis although you see analysis results in notifications. If you deleted an event file, reupload the file.
- Although you see analysis results in notifications, at the time the analysis runs, no valid parameters for the object and dimension values are found. If you deleted a parameter file, reupload the file or set the parameter in the product.

## Analysis does not run per the analysis interval setting

Analysis is configured to run periodically, but does not run per the analysis interval setting.

Check your notifications. The analysis does not run in the following situations:

- For variable analysis, there is no new data and no changes to the parameters. For attribute analysis, there is no new data, no changes to the parameters, and no change to the analysis interval setting.
- There are no event files for the analysis although you see analysis results in notifications. If you deleted an event file, reupload the file.
- Although you see analysis results in notifications, at the time the analysis runs, no valid parameters for the object and dimension values are found. If you deleted a parameter file, reupload the file or set the parameter on the UI.

## Unread notifications remain after you view notifications

After you click the notification icon and view notifications, you still see an alert for unread notifications.

When you click the notification icon, only the latest 20 notifications are shown and are marked as read. Scroll to the end of the notification list and click **load more** to see other notifications. After you view all notifications, the number of unread notifications changes to 0.

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*
*Legal and Intellectual Property Law*
*IBM Japan Ltd.*
*19-21, Nihonbashi-Hakozakicho, Chuo-ku*
*Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

## Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

### Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

### Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

### Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM Online Privacy Statement

IBM Software products, including software as service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user’s name, user name, password, or other personally identifiable information for purposes of session management, authentication, single sign-on configuration or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also likely eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies” and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Printed in USA