

IBM Predictive Maintenance and Quality
Version 2.5.3

Solution Guide



Note

Before using this information and the product it supports, read the information in "Notices" on page 239.

Product Information

This document applies to IBM Predictive Maintenance and Quality Version 2.5.3 and may also apply to subsequent releases.

Licensed Materials - Property of IBM

© **Copyright IBM Corporation 2013, 2016.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Introduction	ix
Chapter 1. What's new	1
What's new in version 2.5.3	1
What's new in version 2.5.1	1
IBM Predictive Maintenance and Quality for Maximo.	1
z BladeCenter Extension support	2
Oracle Database support	2
What's new in version 2.5	2
Feature-based analytics	2
Enhanced integrated analytics	3
Models for IBM Insights Foundation for Energy	3
Quality Early Warning System algorithm for parametric	4
Dashboard application	4
QEWS quality dashboards	5
New Analytics Solutions Foundation version	5
Enhanced QEWS algorithm for Inspection and Warranty.	5
What's new in version 2.0	6
Quality early warning system.	6
Maintenance analytics	6
Top failure predictors	7
Reports	7
Analytics Solutions Foundation	8
Maximo integration	8
Accessibility	8
Chapter 2. Predictive Maintenance and Quality	9
IBM Predictive Maintenance and Quality tasks	10
Identification of the assets, resource types, event types, and measurement types	10
Create a custom application	12
Integration with asset management and manufacturing execution systems	12
IBM Predictive Maintenance and Quality dashboard application.	12
Creating a dashboard	13
Chapter 3. Orchestration	15
Message flows	15
Example of an orchestration XML file.	17
Generic batch orchestration	18
Chapter 4. Master data	23
Master data process	23
File format and location	24
Master data using InfoSphere MDM Collaboration Server	26
IBM Master Data Management Collaboration Server dynamic references	27
Creating a company in IBM InfoSphere MDM Collaboration Server	28
Configuring the IBM InfoSphere MDM Collaboration Server user interface	29
Guidelines for managing data in IBM InfoSphere MDM Collaboration Server	29
Configuring and running data exports	30
Importing metadata into InfoSphere MDM Collaboration Server.	31
Solution XML file	32
IBM Maximo Asset Management	34
How master data is mapped in IBM Maximo Asset Management	34
Mapping master data in IBM Maximo Asset Management	36
Enabling master data loading in realtime mode	38

Importing event data from IBM Maximo Asset Manager	39
Creating a work order service in IBM Maximo Asset Management	39
Configuring work orders in Maximo	40
Mapping work orders for maintenance	48
Chapter 5. Event data	51
How events are processed	51
Event definition	52
Flat file event input.	53
Schema definition of the event format	55
Profile and KPI tables	55
Profile variables	55
KPI tables	56
Profiles	58
Profile calculations	59
Custom calculations	61
Predictive scoring	61
Events and actual, planned, and forecast values	62
Event processing queue	62
Event processing	62
Remove events	64
Configuring solution.xml for the event flow	64
Chapter 6. Quality early warning system use cases	67
Quality inspection	67
Business and technical challenges	68
Defining the quality inspection solution	68
Quality inspection solution details.	69
Results and benefits	74
Warranty	74
Business and technical challenges	76
Defining the warranty solution	76
Warranty solution details.	76
Results and benefits	86
Parametric.	86
Business and technical challenges	88
Defining the parametric solution	88
Parametric solution details	89
Results and benefits	96
Chapter 7. IBM Insights Foundation for Energy	97
Analytical content for the energy and utilities industry	98
Data model changes	100
Orchestration rules	100
Managing the Standard Operating Procedures	102
Configuring Standard Operating Procedures for different activities	105
Defining a Standard Operating Procedure	109
Creating a reference for Standard Operating Procedures	110
Editing a Standard Operating Procedure	111
Submitting a draft Standard Operating Procedure for approval	111
Testing a Standard Operating Procedure	112
Exporting a Standard Operating Procedure	112
Importing a Standard Operating Procedure	112
Reverting to a particular version of a Standard Operating Procedure	113
Viewing a Standard Operating Procedure	113
Chapter 8. Predictive models.	115
The Maintenance predictive model	116
Data understanding	116
Pre-modeling the data	117

Modeling the data	117
Post modeling data manipulation.	118
Evaluation of the model	119
Deployment of the model	120
Recommendations from ADM	120
The Sensor Health predictive model.	121
Data understanding	121
Data preparation	122
Data modeling	124
Evaluation of the model	125
Deployment	126
Recommendations	127
The Top Failure Reason predictive model	127
Understanding the data	127
Preparing the data.	127
Modeling the data.	128
Evaluation	128
Deployment	129
The Feature-based predictive model	129
Input data for training	132
Minimum data requirements	132
Resource sub type level modeling	133
Training job	133
Data preparation	134
Data modeling	134
Deployment	136
Recommendations	137
The Integrated predictive model	137
Input data for training	138
Minimum data requirements	138
Resource sub type level modeling	138
Training job	139
Data preparation	139
Orchestration rules	140
Predictive modeling	140
Deployment	142
Recommendations	143
Chapter 9. Recommendations	145
Preventing scoring for incoming events.	146
Disabling work order creation	146
Chapter 10. Reports and dashboards	147
Site Overview Dashboard	148
Top 10 Contributors dashboard	150
KPI trending report	151
Actual vs plan report.	151
Equipment listing report.	151
Outliers report	152
Recommended actions report	153
Equipment dashboard	153
Equipment profile report	153
Equipment control chart.	154
Equipment run chart	155
Equipment outliers	155
Event type history report	156
Product quality dashboard	156
Defect analysis dashboard	156
Inspection rate analysis	158
Material usage by process crosstab	159

Audit Report	159
Material usage by production batch	160
Maintenance Overview Dashboard	160
Statistical process control reports	163
SPC - Histogram	163
SPC - X Bar R/S Chart	164
Advanced KPI Trend Chart.	164
QEWS quality dashboards	165
Quality dashboard - Inspection	165
Quality dashboard - Inspection Detail History	166
QEWS - Inspection Chart	166
Quality dashboard - Warranty.	167
Quality dashboard - Warranty Detail History.	168
QEWSL - Warranty Chart	168
Quality dashboard - Parametric	169
Quality dashboard - Parametric Detail History	170
QEWSV - Parametric chart	170
TopN Failure Analysis Report	171

Appendix A. Accessibility features 173

Appendix B. The flat file API 175

Master data in the API	175
batch_batch	176
event_code	177
group_dim	177
language	178
location	179
material	180
material_type	180
process	181
product	182
production_batch	182
profile_calculation.	183
resource	184
resource_type	185
source_system	186
supplier	187
tenant	187
Changing the tenant code and name.	188
value_type	188
Metadata in the API	189
event_type	189
measurement_type	189
profile_variable	190
Mandatory profile variables and measurement types	192
Remove master data	193

Appendix C. IBM Cognos Framework Manager model description. 197

IBM Cognos Framework Manager model database layer	197
IBM Cognos Framework Manager model logical layer.	212
IBM Cognos Framework Manager model dimensional layer	213
IBM Cognos Framework Manager model security	213
Query mode.	213
Using Compatible Query Mode to see real-time data	213

Appendix D. IBM Predictive Maintenance and Quality Artifacts 215

Data model	215
IBM InfoSphere Master Data Management Collaboration Server file	215
IBM Integration Bus and ESB artifacts	215

Sample master data, event data, and QEWS data files	217
IBM SPSS artifacts	218
IBM Cognos Business Intelligence Artifacts	224
Appendix E. Troubleshooting	231
Troubleshooting resources	231
Support Portal	231
Service requests	232
Fix Central	232
Knowledge bases	232
Log files	233
Unresolved library error on import of PMQDancingCharts or PMQMasterDDLGenerator	234
SPSS job invocation from the orchestration message flow fails	235
Performance tuning guidelines	235
Deadlock errors happen when parallel processing is enabled	235
Event processing performance	236
Troubleshooting reports	237
Audit Report fails with error DMB-ECB-0088 A DMB cube build limit has been exceeded	237
Notices	239
Index	243

Introduction

The IBM® Predictive Maintenance and Quality solution uses data from multiple sources to give you the information to make informed operational, maintenance, or repair decisions.

IBM Predictive Maintenance and Quality provides you with operational intelligence data, which enables you to perform the following tasks:

- Understand, monitor, predict, and control product and process variability.
- Perform in-depth root cause failure analysis.
- Identify incorrect operating practices.
- Enhance equipment and process diagnostics capabilities.

It also provides you with asset performance management capabilities that help you to achieve these goals:

- Have forward visibility into equipment and process performance.
- Increase asset uptime.
- Identify safety issues.
- Identify improper maintenance procedures.
- Optimize maintenance intervals and procedures.

Audience

This information is intended to provide users with an understanding of how the IBM Predictive Maintenance and Quality solution works. It is designed to help people who are planning to implement IBM Predictive Maintenance and Quality know what tasks are involved.

Finding information

To find documentation on the web, including all translated documentation, access IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. Some of the components included in the IBM Predictive Maintenance and Quality solution have accessibility features. For more information, see Appendix A, “Accessibility features,” on page 173.

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

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The

development, release, and timing of features or functionality remain at the sole discretion of IBM.

Chapter 1. What's new

This section describes new and changed features in IBM Predictive Maintenance and Quality.

What's new in version 2.5.3

The following new features are available in IBM Predictive Maintenance and Quality 2.5.3.

Solution installer

A new solution installer is available for installations on 64-bit Red Hat Enterprise Linux Server Edition version 6 (x86_64 with update 5, 6, or 7) operating systems. The solution installer provides a new user interface for flexible deployment topology creation. This installer provides a simplified set of prerequisite libraries for loading the installer. It also includes pre-built templates and other enhancements that help ensure a smooth Predictive Maintenance and Quality custom or silent installation into an enterprise environment.

For more information, see the following topics:

- Install the Predictive Maintenance and Quality servers
- Install the Predictive Maintenance and Quality server artifacts

Other updates

IBM Predictive Maintenance and Quality 2.5.3 includes recent supporting program updates and fixes available at the time of release.

Deprecation notices

IBM Predictive Maintenance and Quality 2.5.3 will be the last version to include IBM Infosphere Master Data Management Collaborative Edition. Master Data Management CE is not installed by default. For information about installing Master Data Management CE, see the installation instructions (http://www.ibm.com/support/knowledgecenter/SSWSR9_11.5.0/com.ibm.swg.im.mdmhs.common.install.nav.doc/topics/install_cont.html).

What's new in version 2.5.1

There are several new features in IBM Predictive Maintenance and Quality for this release.

IBM Predictive Maintenance and Quality for Maximo

For licensees of the IBM Maximo® solution, IBM Predictive Maintenance and Quality offers Predictive Maintenance and Quality for Maximo.

Predictive Maintenance and Quality for Maximo is a full-featured analytics extension of your organization's current Maximo investment. It provides IBM Maximo users with the full functionality of Predictive Maintenance and Quality with special license terms.

Predictive Maintenance and Quality and Maximo are seamlessly integrated through a prebuilt adapter. Predictive Maintenance and Quality for Maximo supports the following capabilities:

- Master data loading of asset information that is already defined in Maximo.
- Predictive analysis of maintenance data that is supplied from Maximo and external sources in real time or batch.
- Automatic creation of analytics-driven maintenance work orders in Maximo based on maintenance recommendations from Predictive Maintenance and Quality.

z BladeCenter Extension support

IBM z™ BladeCenter Extension (zBX) is the infrastructure that extends IBM z Systems™ governance and management capabilities across a set of integrated, fit-for-purpose IBM System x compute elements in IBM z Systems.

zBX expands the z Systems™ portfolio to applications that run on AIX®, Linux on System x, and Microsoft Windows. IBM Predictive Maintenance and Quality 2.5.1 leverages its x86 Linux support to run in this environment.

Oracle Database support

IBM Predictive Maintenance and Quality now supports Oracle Database v12c as an alternative database to store Predictive Maintenance and Quality data.

Predictive Maintenance and Quality 2.5.1 contains updates to the Analytic Solutions Foundation, Predictive Maintenance and Quality data schema, adapters, and models to provide seamless support to Oracle Database for Predictive Maintenance and Quality-specific functions.

What's new in version 2.5

There are several new and changed features in IBM Predictive Maintenance and Quality for this release.

Feature-based analytics

IBM Predictive Maintenance and Quality Feature-Based Analytics is an improvement to Sensor Analytics in Predictive Maintenance and Quality 2.0. Feature-Based Analytics analyzes historical failures and planned or unplanned (breakdowns) maintenance orders that are generated in IBM Maximo. It predicts a resource's health score and optimal maintenance period, and recommends inspection or changes to maintenance schedules.

Predictive Maintenance and Quality Feature-Based Analytics has the following features:

- A framework to configure domain-specific user-defined features and raw sensor events and KPI data. User-defined features are different transformations that use a combination of one or more raw sensor events and static and lifetime parameters across different data types.
- Improved Predictive Maintenance and Quality performance. Predictive Maintenance and Quality makes a single call between IBM Integration Bus and IBM SPSS Modeler for invocation calls. A single call is now made to SPSS for the health score, days to maintenance, and recommendation calculations. In Predictive Maintenance and Quality 2.0 Sensor Analytics, separate calls are made.

- A new design that works on the same features to build models at resource sub type level segments. The new design can produce effective insights even during the period when new equipment is yet to achieve optimal maturity for effective predictive modeling. This capability provides a quicker return on investment for similar equipment, or for cases like transformers that fail only once in a lifetime, and therefore are incapable of equipment level predictions.
- A seamless integration, comparison, and substitution of customized analytics in a format that is compatible with existing Sensor Analytics and Maintenance Analytics, and other similar analytics implementations.

Related concepts:

“The Feature-based predictive model” on page 129

The Feature-based predictive model produces a resource's predicted health score and optimal maintenance period, and recommends inspection or changes to maintenance schedules.

Enhanced integrated analytics

IBM Predictive Maintenance and Quality Integration Analytics provide a seamless and plug-and-play based integration, substitution, and comparison across different types of analytics, including maintenance, sensor, and feature-based analytics.

Integration Analytics predicts a machine's health and optimal maintenance period, and provides maintenance recommendations. The recommendations provide users with composite actionable insights.

Predictive Maintenance and Quality integrated analytics has the following features:

- Combines feature-based analytics scores and scores from sensor and maintenance analytics from Predictive Maintenance and Quality 2.0.
- Can be used to combine Text Analytics or other custom and compatible analytics formats that users might use to predict a machine's health and maintenance recommendations.
- Intelligent prediction of a machine's health, optimal days to maintenance, and recommendations based on models with predictions customized for each resource.
- Supports resource sub type level models like in feature-based analytics, which can give recommendations for a new equipment based on resource sub type level characteristics.
- Automatic training and refresh of prediction models at preset intervals.
- Option to manually refresh the deployed model for the entire batch of resources with a single click during ad hoc or sudden changes in sensor data.
- Automatic filtering of resources with scarce data to guard against model generation and refresh failures.

Related concepts:

“The Integrated predictive model” on page 137

The Integrated predictive model produces a predicted health score and days to maintenance for each asset or process at a site. The health score is used to determine an asset's performance.

Models for IBM Insights Foundation for Energy

IBM Predictive Maintenance and Quality provides models for IBM Insights Foundation for Energy.

The Predictive Maintenance and Quality solution provides health scores of assets that are commissioned in Insights Foundation for Energy. Insights Foundation for Energy computes relevant features for the energy and utilities industry. The features are incorporated within Feature-Based Analytics in Predictive Maintenance and Quality.

The following features are supported:

- Specific feature computations to support repair-rehab-replace. The computations are incorporated through Analytics Solutions Foundation and IBM SPSS batch jobs.
- Features for substation transformers, distribution transformers, poles, and cables are covered.
- Current and projected aging for distribution transformers are provided.

Related concepts:

Chapter 7, “IBM Insights Foundation for Energy,” on page 97

The IBM Predictive Maintenance and Quality solution provides health scores of assets that are commissioned in IBM Insights Foundation for Energy.

Quality Early Warning System algorithm for parametric

IBM Predictive Maintenance and Quality provides a Quality Early Warning System for Variables (QEWSV) algorithm to detect unfavorable changes in variable-type data.

The QEWSV algorithm analyzes variable data and detects unfavorable changes by plotting variable value plots and evidence charts. The QEWS parametric use case can generate plots for any sub use case, including Process Resource Validation, Resource Validation, Product Validation, Material Validation, and Location Suitability.

Related concepts:

“Parametric” on page 86

The quality early warning system (QEWS) parametric use case in IBM Predictive Maintenance and Quality detects unfavorable changes in variable-type data, and provides information that facilitates diagnostics and alarm prioritization.

Dashboard application

IBM Predictive Maintenance and Quality provides a dashboard application for monitoring real-time event data for a selected resource and profile.

The Predictive Maintenance and Quality dashboard application provides a single interface where users can access IBM Cognos Business Intelligence reports, access the IBM Analytics Solutions Foundation, and view live profile values for a particular resource.

The dashboard application supports the following features.

- No limit to the number of dashboards users can create.
- No limit to the number of charts a dashboard can contain.
- Users can create, delete, save, and reorder charts.
- Users can subscribe to live data by selecting a resource and profile.
- Controlled access to the Analytics Solutions Foundation interface.

Related concepts:

“IBM Predictive Maintenance and Quality dashboard application” on page 12
The Predictive Maintenance and Quality dashboard application provides a single interface where users can access IBM Cognos® Business Intelligence reports, access the IBM Analytics Solutions Foundation, and view live profile values for a particular resource.

QEWS quality dashboards

IBM Predictive Maintenance and Quality provides quality dashboards that give users an instant overview of products with the help of stoplights.

Business needs directly influence stoplight computation policies, and can create rules that affect stoplight evaluation.

The quality dashboards have the following features.

- Three sets of quality dashboards for the QEWS inspection, warranty, and parametric use cases.
- The status of a product can be analyzed across the product hierarchy for up to five levels.
- Quality dashboards are linked to detail reports that show various threshold values that are associated with the products.
- QEWS charts are linked to the detailed reports to provide seamless navigation and continued analysis from the overview to the details.

Related concepts:

“QEWS quality dashboards” on page 165

The QEWS quality dashboards show data for the QEWS quality inspection, warranty, and parametric use cases.

New Analytics Solutions Foundation version

A new version of Analytics Solutions Foundation is included with IBM Predictive Maintenance and Quality.

Analytics Solutions Foundation 1.5 is packaged with Predictive Maintenance and Quality. Version 1.5 has the following features.

- Foundation user interface
- BPM adapter, Scoring adapter, and Event Validation adapter
- Custom surrogate key column
- Support for BIGINT data type
- Last updated time stamp generation

The Analytics Solutions Foundation Programming Guide 1.5 is available as a Techdoc. To download the programming guide, see the Techdoc titled Analytics Solutions Foundation Programming Guide 1.5 (<http://www.ibm.com/support/docview.wss?uid=swg27045086>).

Enhanced QEWS algorithm for Inspection and Warranty

The Quality Early Warning System (QEWS) algorithm is enhanced for the inspection and warranty use cases.

QEWS inspection identifies an instance of a given manufacturing process deviation leading to quality impacts, as captured by applying statistical procedures on the observed failure rate during inspections, earlier than many of the conventional

quality inspection statistical process control systems, such as p-chart. The enhancements to QEWS warranty include:

- rundate support for Inspection
- support for storing historical run results
- QEWS charts that are produced by the Rapidly Adaptive Visualization Engine (RAVE) in IBM Cognos Business Intelligence
- inspection and parameter loading that is aligned with standard Predictive Maintenance and Quality event load mechanism
- file-based triggering introduced for QEWS Invocation Adapter

QEWS warranty identifies instances of accelerated wear or replacement rate for a product that is under service, due to in-service conditions, earlier than many of the conventional statistical and non-statistical procedures can indicate. The enhancements to QEWS warranty include:

- improved rundate support
- parallel support for multiple use cases (Sales, Manufacturing, Production)
- QEWS charts that are produced by the Rapidly Adaptive Visualization Engine (RAVE) in Cognos Business Intelligence
- parameter loading that is aligned with standard Predictive Maintenance and Quality event load mechanism

Related concepts:

Chapter 6, “Quality early warning system use cases,” on page 67

The quality early warning system (QEWS) in IBM Predictive Maintenance and Quality 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.

What's new in version 2.0

There are several new and changed features in IBM Predictive Maintenance and Quality for this release.

Quality early warning system

The quality early warning system (QEWS) uses the advanced analytics, visualization, and workflow in IBM Predictive Maintenance and Quality to detect quality problems earlier and more definitively.

QEWS monitors large amounts of quality data automatically, with earlier alerts, definitive alerts, and intelligent prioritization. For more information about the QEWS, see Chapter 6, “Quality early warning system use cases,” on page 67.

Maintenance analytics

IBM Predictive Maintenance and Quality maintenance analytics predicts the optimal conditions for a resource by analyzing historical, planned, and breakdown maintenance work orders. The analysis is used to recommend customized changes to the maintenance schedule of the resource.

Predictive Maintenance and Quality maintenance analytics has the following features:

- Advanced modeling that draws maintenance insights from intermittent and censored breakdown and scheduled maintenance events.
- Customized analytics in a format that is compatible with other analytics software. Compatibility with other analytics software enables seamless integration, comparison, and substitution between Predictive Maintenance and Quality and other statistical products.
- Works independently of sensor data. Predictive Maintenance and Quality can produce effective insights before sensor data is at optimal maturity for effective predictive modeling. This capability provides quicker return on investment.
- Intelligent prediction of a machinery health and recommendation that is based on models with predictions customized for each resource.
- Automatic training and refresh of prediction models at preset intervals.
- Option to manually refresh the deployed model during ad hoc or when a sudden change in sensor data occurs.
- Automatic filtering of resources with scarce data for predictive model generation.
- Can be used to combine text analytics, or other custom and compatible analytics format to predict machinery health and maintenance recommendations.

Top failure predictors

This feature helps you to understand the top reasons that a resource fails. You can use the supplied statistical process control (SPC) charts to perform root cause analysis that leads to pattern discovery.

IBM Predictive Maintenance and Quality top failure predictors has the following features:

- Ability to analyze and discover the top percentile or number of parameters that predict the failure or optimal health of a resource.
- Ability to drill through for a selected resource to see a detailed analysis of its patterns and anomaly detection.
- Custom analytics with any number of parameters or profiles for a resource.
- Ability to perform predictor importance analysis on custom profiles, features, and calculations by creating new custom profiles. For example, you can create profiles for cumulative humidity instead of absolute humidity.

For more information about the TopN Failure Analysis Report and the SPC reports, see Chapter 10, “Reports and dashboards,” on page 147.

Reports

IBM Predictive Maintenance and Quality provides new reports for statistical process control and the quality early warning system (QEWS). There is a new key performance indicator (KPI) trend chart. Health score reporting is improved.

The Advanced KPI Trend chart displays separate charts for multiple profiles across multiple resources. The TopN Failure Analysis report shows the profiles that contribute to the failure of a resource.

The Maintenance Overview Report shows the sensor health score, maintenance health score, and the integrated health score for the resources at a location.

Statistical process control charts

The following new reports analyze statistical process control:

- SPC - Histogram
- SPC - X Bar R/S Chart

Quality early warning system charts

The following new reports support QEWS:

- QEWS - Inspection Chart
- QEWSL - Warranty Chart

For more information, see Chapter 10, “Reports and dashboards,” on page 147.

Analytics Solutions Foundation

You can use IBM Analytics Solutions Foundation to extend or modify IBM Predictive Maintenance and Quality.

Analytics Solutions Foundation is an alternative to using the flat file application programming interface (API) to extend the Predictive Maintenance and Quality solution. Analytics Solutions Foundation helps you to define orchestrations without writing code to integrate the API.

Maximo integration

IBM Predictive Maintenance and Quality and IBM Maximo are seamlessly integrated.

The integration with Maximo includes the following features:

- Support for updating an existing maintenance work order in Maximo with the maintenance recommendation from Predictive Maintenance and Quality.
- Support for processing maintenance Maximo work orders in batch and real time.
- Support for master data loading in real time.

Accessibility

The reports in IBM Predictive Maintenance and Quality are accessible.

For more information, see Appendix A, “Accessibility features,” on page 173.

Chapter 2. Predictive Maintenance and Quality

With IBM Predictive Maintenance and Quality, you can monitor, analyze, and report on information that is gathered from devices. In addition, recommendations for actions can be generated by Predictive Maintenance and Quality.

IBM Predictive Maintenance and Quality is an integrated solution that you can use to perform the following tasks:

- Predict the failure of an instrumented asset so that you can prevent costly unexpected downtime.
- Make adjustments to predictive maintenance schedules and tasks to reduce repair costs and minimize downtime.
- Quickly mine maintenance logs to determine the most effective repair procedures and maintenance cycles.
- Identify the root cause of asset failure faster so that you can take corrective actions.
- Identify quality and reliability issues definitively and in a timely way.

Instrumented assets generate data such as device ID, timestamp, temperature, and status code. This data can be collected and used with maintenance records and other data in models that predict when an asset is likely to fail.

Examples of instrumented assets are manufacturing equipment, mining equipment, drilling equipment, farming equipment, security equipment, cars, trucks, trains, helicopters, engines, cranes, oil platforms, and wind turbines.

For example, an oil refinery is a system that combines thousands of interlocking pieces. It is critical that such a system is able to work safely and efficiently. You can use IBM Predictive Maintenance and Quality to monitor and track the lifecycle of each piece of the refinery, such as each pipe, pump, compressor, valve, furnace, turbine, tank, heat exchange unit, and boiler. Reports give you the information to ensure that you have the necessary parts available and can schedule repairs during periods of inactivity.

Predictive maintenance

In predictive maintenance, you look for patterns in the usage and environmental information for equipment that correlate with failures that take place. This information is used to create predictive models to score incoming new data. You can predict the likelihood of failure. Scores are generated from this information that give you an indication of the health of the piece of equipment. In addition, key performance indicators (KPIs) are collected, which are used for reporting. KPIs help you to identify assets that do not conform to normal patterns of behavior. You can define rules to generate recommendations when a piece of equipment is identified as having a high probability of failure. Recommendations can be fed into other systems so that people are alerted to them automatically.

Predictive quality in manufacturing

Past operations data, environmental data, and historical defect data can be used to identify the causes of elevated defect rates. This information is used in predictive models, so that when incoming data is fed into the models, you can predict the

likely defect rates. The predicted values are then used for analysis and reporting and to drive recommendations such as modification to inspection patterns, or recalibration of machinery. Scoring can be done on a near real-time basis.

Predictive Maintenance and Quality can also detect quality and reliability problems faster than traditional techniques.

IBM Predictive Maintenance and Quality tasks

You must configure your IBM Predictive Maintenance and Quality application before the application can be deployed to users.

The following tasks are necessary to configure IBM Predictive Maintenance and Quality:

- Identify the assets, resource types, their event types, and measurements.
- Load the master data. Master data supplies IBM Predictive Maintenance and Quality with information about the context in which events occur, for example, location of a resource or event, definition of a material or production process.
- Load the event data. The event data is data that you want to measure about an event. Data comes from many sources, and it must be transformed into a format that can be used by IBM Predictive Maintenance and Quality.
- Configure the event types, measurement types, and profile variables. Set up the types of measurement that must be done, and the key performance indicators (KPIs) that must be calculated from these measurements. Profiles are a condensed history of resources that help to speed up scoring.
- Configure the predictive models. Run the historical data through the modeler to determine which values are needed. You can then refine the model so that it gives you accurate predictions and generates scores.
- Define rules that determine what actions happen when a score threshold is breached.
- Configure the reports and dashboards that the user sees. Reports and dashboards can be customized, and new ones can be created.

Identification of the assets, resource types, event types, and measurement types

Before you deploy an IBM Predictive Maintenance and Quality application, identify the assets and the information that you want to monitor.

To establish what data is required, and what preparation must be done, ask the following questions.

- Which assets must be monitored, and why?
- What events do you want to monitor for those assets?
- What measurements do you want to capture for the events?

Resource types

The two supported resource types are asset and agent. An asset is a piece of equipment that is used in the production process. An agent is the operator of the equipment. When you define resources, you can use the resource subtype field to identify specific groups of assets or agents.

The following table shows some sample event types in the data model.

Table 1. Sample event types in the data model

Event type code	Event type name
ALARM	Alarm
WARNING	Warning
SYSTEM CHECK	System Check
MEASUREMENT	Measurement
RECOMMENDED	Recommended Actions
FAILURE	Failure
REPAIR	Repair

The following table shows some sample measurement types in the data model.

Table 2. Sample measurement types in the data model

Measurement type code	Measurement type name
RECOMMENDED	Recommend Action
RPM	RPM
FAIL	Incident Count
INSP	Inspection Count
LUBE	Lube Count
OPHR	Operating Hours
PRS1	Pressure 1
PRS2	Pressure 2
PRS3	Pressure 3
R_B1	Replace Ball Bearing Count
R_F1	Replace Filter Count
RELH	Relative Humidity
REPT	Repair Time
REPX	Repair Text
TEMP	Ambient Temperature
Z_AC	High Temperature/ Humidity Warning Count
Z_FF	Latent Defect
Z_PF	Probability of Failure
Z_TH	High Temperature/ Humidity Count
OPRI	Operating Hours at Inception
REPC	Repair Count
MTBF	MTBF
MTTR	MTTR
OPRD	Operating Hours Delta

Create a custom application

You can create a custom IBM Predictive Maintenance and Quality application by creating custom IBM Integration Bus flows, IBM Cognos Business Intelligence reports and dashboards, or predictive models.

The following list describe the high-level tasks you can take to create a custom application.

- Customize or create new predictive models by using IBM SPSS® Modeler.
- Create new business rules by using IBM Analytical Decision Management.
- Create new flows that interface with external systems by using IBM Integration Bus.
- Customize scoring during event processing by using IBM Integration Bus.
- Customize or create the message flows to orchestrate the activities by using IBM Integration Bus.
- Customize or create new reports by using IBM Cognos Report Studio.
- Modify the metadata for the reports by using IBM Cognos Framework Manager.

Sample files, model files, and other content is supplied to help you to configure IBM Predictive Maintenance and Quality for the needs of your business. For more information, see Appendix D, “IBM Predictive Maintenance and Quality Artifacts,” on page 215.

Integration with asset management and manufacturing execution systems

Asset management and manufacturing execution systems are an important source of master data and event data. You can feed recommendations and forecasts produced by IBM Predictive Maintenance and Quality into these systems to close the loop and perform action.

Predictive Maintenance and Quality can create work orders in IBM Maximo Asset Management based on recommendations from predictive scoring and decision management. Predictive Maintenance and Quality contains the APIs for integration with these systems and technology for building connectors to the systems. Predictive Maintenance and Quality includes a prebuilt adapter for integration with Maximo.

IBM Maximo is not installed as part of IBM Predictive Maintenance and Quality. If required, it must be purchased separately. However, IBM Predictive Maintenance and Quality contains adapters for IBM Maximo, which allow data integration.

IBM Predictive Maintenance and Quality dashboard application

The Predictive Maintenance and Quality dashboard application provides a single interface where users can access IBM Cognos Business Intelligence reports, access the IBM Analytics Solutions Foundation, and view live profile values for a particular resource.

Users access the Predictive Maintenance and Quality dashboard application by typing the following URL.

```
http://bi_node_name:port_number/PMQDashboard/pmqui
```

bi_node_name is the name or IP address for the BI node computer. Consult your system administrator to obtain the BI node computer name and port number.

IBM Cognos Business Intelligence reports

The **Reports** link in the dashboard application banner gives users access to IBM Cognos BI content. The link opens IBM Cognos Connection, and users can view available reports.

IBM Analytics Solutions Foundation

The **Foundation** link in the dashboard application banner gives users that belong to the Administrator group access to the IBM Analytics Solutions Foundation interface. In this interface, users can author orchestration and solution definition XML files.

Note: The IBM Analytics Solutions Foundation Developer Guide is available from the customer support portal. To access it, see the IBM Analytics Solutions Foundation Developer Guide (<http://www.ibm.com/support/docview.wss?uid=swg27045826>).

Dashboards

The **Dashboard** link in the dashboard application banner gives users the ability to create, save, and delete dashboards that contain one or more charts. Users configure each chart to display live data for a profile of a selected resource.

Related information:

Chapter 10, "Reports and dashboards," on page 147

You can customize and extend the reports and dashboards that are supplied with IBM Predictive Maintenance and Quality. You can also design your own reports and dashboards and add them to the menu.

Creating a dashboard

Create a dashboard to show live data in a chart for a selected profile of a resource.

About this task

The following table describes the tasks that you can perform when you create a dashboard.

Table 3. Dashboard tasks








Icon	Task
	Save dashboard
	Edit dashboard
	Add chart
	Configure chart
	Delete chart
	Move chart up or down

Table 3. Dashboard tasks (continued)

Icon	Task
	Delete dashboard

There is no limit to the number of dashboards that you can create, and there is no limit to the number of charts that a dashboard can contain.


When you work in a dashboard, two modes are available. A dashboard enters the view mode when you save the dashboard. From the view mode, a dashboard enters the edit mode when you click the **Edit dashboard** icon.

When you configure a chart, the chart can display data for only one profile of a selected resource. When you select a resource, you are shown a list of corresponding profiles to choose from. Resources are categorized under Location and Resource sub types to help you search for the resource that you want. You choose a profile by subscribing to it.

After you subscribe to a profile, if you receive a subscription failed message, ask your administrator to check the following possibilities for the failure.

- The Predictive Maintenance and Quality dashboard WebSphere Application Server cannot connect to the IBM Integration Bus (IIB).
- The IIB server is offline.
- The IIB server is incorrectly configured in the application server.

After you configure a chart, live data for the selected profile of a resource is displayed on the chart. At the most, 15 points are displayed on maps. Chart values are displayed by using the first in, first out concept. New values are pushed at the back, and old values are removed from the front.

Tip: If you receive a message that states that no data is available for the selected resource and profile, it might be because there is no data reported for that resource and profile. Otherwise, click the Refresh icon  to try retrieving the data again.

Procedure

1. Log in to the Predictive Maintenance and Quality dashboard application.
2. In the navigation pane, click the **Create a new dashboard** link, and in the work area, in the **Dashboard Name** box, type the name of the dashboard that you want to create.
3. Add a chart to the dashboard.
4. Configure the chart.
 - a. In the Chart Configuration window, select a resource.
 - b. For the selected resource, click the **Profiles** menu and choose a profile.
 - c. Click **Subscribe**.
5. To add more charts to the dashboard, repeat steps 3 and 4.
6. Save the dashboard.

Chapter 3. Orchestration

Orchestration is the process that ties activities in IBM Predictive Maintenance and Quality together.

Message flows

Orchestration is achieved with message flows in IBM Integration Bus.

The following activities can be tied together:

- Acquiring and storing data
- Aggregating data
- Running predictive models
- Feeding data back to external systems or starting external processes

Message flows are supplied with IBM Predictive Maintenance and Quality and must be customized with IBM Integration Bus. The message flows are organized into the following applications:

- PMQEventLoad
- PMQMasterDataLoad
- PMQMaximoOutboundIntegration
- PMQMaintenance
- PMQModelTraining
- PMQQEWSInspection
- PMQQEWSIntegration
- PMQQEWSWarranty
- PMQTopNFailure

For more information about developing message flows, see IBM Integration Bus Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSMKHH_9.0.0/com.ibm.etools.mft.doc/bi12005_.htm).

By default, IBM Integration Bus is installed in advanced mode. Advanced mode is the correct mode to use for full functionality.

The following examples describe how orchestration is used in IBM Predictive Maintenance and Quality.

Orchestration example: Load real-time event data

This orchestration example is similar to the message flow used to load batch event data.

1. Incoming equipment measurement data is provided through real-time connectivity.
2. A map must be defined in IBM Integration Bus to describe the transformation of incoming data into the IBM Predictive Maintenance and Quality event structure.
3. Incoming business keys are converted to internal integer surrogate keys.

4. Event data is written to the datastore.
5. Event data is aggregated. Profile and key performance indicator (KPI) data is written to the datastore.

Orchestration example: Load batch event data

The following steps take place when batch event data is loaded into IBM Predictive Maintenance and Quality.

1. Incoming measurement data is loaded from a file.
2. The file system is polled for new incoming data.
3. A map that is defined in IBM Integration Bus describes the transformation of incoming data into the IBM Predictive Maintenance and Quality structure.
4. Incoming business keys are converted to internal integer surrogate keys.
5. Event data is written to the datastore.
6. Event data is aggregated. Profile and key performance indicator (KPI) data is written to the datastore.

Orchestration example: Score event data

The following steps take place when event data is scored.

1. New input triggers scoring. For example, to recalculate the health score if a new measurement is reported, then that measurement is processed and the health score is recalculated.
2. A map that is defined in IBM Integration Bus describes the transformation of the data into the model structure.
3. The predictive model is invoked through a web services interface.
4. A map that is defined in IBM Integration Bus describes the transformation of model outputs to the event structure.
5. Model outputs are written as new events.
6. As with external events, model output events can be aggregated and stored on the profile and as KPIs.

For more information about scoring predictive models, and the triggers to score models, see “Predictive scoring” on page 61.

Orchestration example: Apply business rules to data

The following steps take place when business rules are applied.

1. New input triggers evaluation of business rules.
2. A map that is defined in IBM Integration Bus describes the transformation of the data into the model structure.
3. IBM Analytical Decision Management Model is invoked through a web services interface.
4. A map that is defined in IBM Integration Bus describes transformation of model outputs to the event structure.
5. Model outputs are written as new events.
6. As with external events, model output events can be aggregated and stored on the profile and as KPIs.

Orchestration example: Write-back of data

The following steps take place when write-back of data to an external process occurs.

1. Creation of an event triggers the requirement to start an external process.
2. A map that is defined in IBM Integration Bus describes the transformation of the data to the structure of an external web service.
3. The external web service is called.

Example of an orchestration XML file

An example file, `inspection.xml`, demonstrates the purpose and structure of an orchestration file.

Each orchestration flow can be defined in a separate XML file. The file defines the behavior of the orchestration steps. A mapping determines the orchestrations to be performed for an event with an event orchestration key code.

In this example scenario, there are two kinds of events: production and inspection. Therefore, there are two event orchestration key codes, one for each type of event.

The example file “`inspection.xml`” on page 18 determines the orchestration for an inspection event.

Description

The first part of the file `inspection.xml` lists the event type, the adapter class, and configuration that is required for the particular class of adapter:

- `<event_orchestration_mapping>`
The type of event is defined as an inspection.
- `<adapter_class>`
The adapter class that will be executed, in this case `ProfileAdapter`, is called in the step.
- `<adapter_configuration>`
The profile adapter requires configuration to determine how observations with a specific measurement type will update specific profile tables.

The remainder of the file specifies how two specific profiles will be updated, depending on whether the measurement type has a value of `INSPECT` or `FAIL`:

- `<observation_profile_update>`
If the measurement type has a value of `INSPECT`
`<profile_update_action>` The `PRODUCT_KPI` table is updated with the shared calculation of `Product_KPI_Inspect_count`. This calculation produces the value for the number of days when an inspection has taken place.
- `<observation_profile_update>`
If the measurement type has a value of `FAIL`
`<profile_update_action>` The `PRODUCT_KPI` table is updated with the shared calculation of `PRODUCT_KPI_FAIL_COUNT`. This calculation produces the value for the number of times an asset has failed.

inspection.xml

The file inspection.xml contains the following code:

```
<event_orchestration_mapping>
  <event_orchestration_key_cd>inspection</event_orchestration_key_cd>
  <orchestration_cd>pmq.inspection</orchestration_cd>
</event_orchestration_mapping>

<orchestration>
  <orchestration_cd>pmq.inspection</orchestration_cd>
  <step>
    <adapter_class>com.ibm.analytics.foundation.adapter.profile.ProfileAdapter</adapter_class>
    <adapter_configuration xsi:type="ns3:profile_adapter_configuration">
      <observation_profile_update>
        <observation_selector table_cd="EVENT_OBSERVATION">
          <observation_field_value>
            <field_name>MEASUREMENT_TYPE_CD</field_name>
          </observation_field_value>
          <value>INSPECT</value>
        </observation_selector>

        <profile_update_action>
          <profile_row_selector>
            <shared_selector_cd>PRODUCT_KPI</shared_selector_cd>
          </profile_row_selector>
          <shared_calculation_invocation_group_cd>PRODUCT_KPI_INSPECT_COUNT
          </shared_calculation_invocation_group_cd>
        </profile_update_action>
      </observation_profile_update>

      <observation_profile_update>
        <observation_selector table_cd="EVENT_OBSERVATION">
          <observation_field_value>
            <field_name>MEASUREMENT_TYPE_CD</field_name>
          </observation_field_value>
          <value>FAIL</value>
        </observation_selector>
        <profile_update_action>
          <profile_row_selector>
            <shared_selector_cd>PRODUCT_KPI</shared_selector_cd>
          </profile_row_selector>
          <shared_calculation_invocation_group_cd>
PRODUCT_KPI_FAIL_COUNT</shared_calculation_invocation_group_cd>
          </profile_update_action>
        </observation_profile_update>
      </adapter_configuration>
    </step>
  </orchestration>
```

Generic batch orchestration

Generic batch orchestration provides capabilities to run a scheduler flow and invoke any IBM SPSS batch job by taking inputs from a configurable XML file instead of developing separate message flows for a specific use case.

Generic batch orchestration also provides the flexibility of changing the scheduled time of a flow and web service input parameters through an XML file, without changing message flows.

In generic batch orchestration, an orchestration XML file is used to store all configurations. Message flows read the XML file at run time from the properties folder on the Enterprise Service Bus (ESB) node.

The following capabilities are available in generic batch orchestration.

AutoTrigger

AutoTrigger is used to automatically trigger scheduler flows. The AutoTrigger flow creates a TimeoutRequest message according to the scheduler configurations in the XML file, and places the message on the queue that is specified in the scheduler configuration. AutoTrigger can accept any changes that are made to scheduler configurations.

The queue to which the AutoTrigger flow places the TimeoutRequest message can either be SPSSJobIntegration.msgflow or any other custom flow that is configured to run at a scheduled time. To run at the scheduled time, SPSSJobIntegration.msgflow or a custom flow must contain an MQInput node, TimeoutControl node, and a TimeoutNotification node that is configured to match the Identifier and queue name configurations in the XML file.

The following figure shows the parameters that are required for a scheduler configuration.

batch	Orchestration for Parametric
orchestration	Parametric
Identifier	Parametric
scheduler	
scheduled_time	00:00:00
queue_name	PMQ.QEWS.PTIMER.IN
duration_in_days	1

Figure 1. Required parameters for a scheduler configuration

The first trigger of the flow occurs on the date of the flow deployment or flow restart at the time specified (<scheduled_time></scheduled_time>) in the batch orchestration XML file. The trigger repeats at regular intervals, in days (<duration_in_days></duration_in_days>), as specified in the batch orchestration XML file.

If a change is made to **duration_in_days**, the change takes effect from the second run that was previously configured. For example, **duration_in_days** is set to 3, and with this value, the next time the flow is supposed to run is 2014-12-09. If **duration_in_days** is changed to 2, the change takes effect only after 2014-12-09. If you want the change to immediately take effect, you must restart the flow.

The following figure shows an example of the Timer flow.

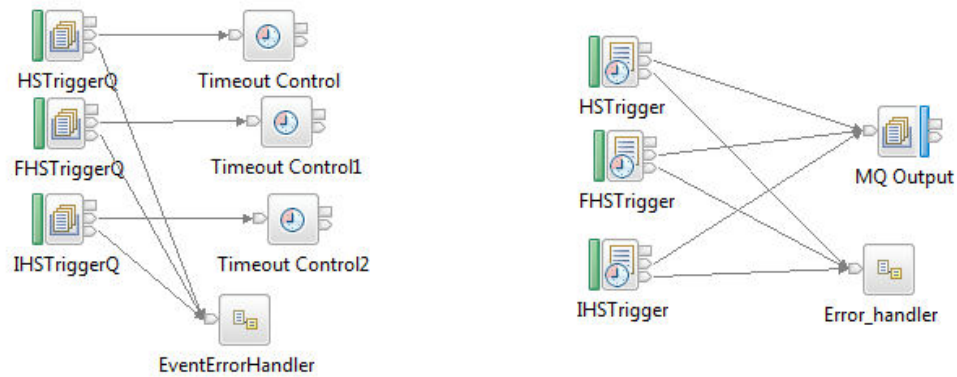


Figure 2. Timer flow example

SPSSJobIntegration

The SPSSJobIntegration flow is used to invoke the IBM SPSS SubmitJobWithOptions web service to trigger an SPSS job with parameters. The SPSSJobIntegration flow picks up parameters, the web service end-point URL, the SPSS JobLocation URL, and **notificationEnabled** field values from the XML file.

Parameters that are used in the web service can either be static parameters, which are values that are predefined in the XML file, or dynamic parameters, which are values that are expected to come from a data preparation flow. The data preparation flow is specific to a use case, or is a combination of both static and dynamic parameters. There can also be cases where there are no parameters that are required.

The **type** field in the XML file specifies whether the parameter is static or dynamic. The parameter name is configured in the **name** field. For static parameters, the parameter value is specified in the **value** field. For dynamic parameters, the value is taken from the input message in the data preparation flow, which is specific to a use case. In this case, the **value** field in the XML file contains the input message field name, from which the dynamic value is mapped.

For dynamic field mapping, the request from the data preparation flow must have **Request** as the parent XML tag name, and child elements must contain the parameters. This request must also contain **Identifier** as one of the child elements, which is used by SPSSJobIntegration to identify which set of parameters to use for the particular use case.

The following code is a sample request from a data preparation flow. StartDate is one of the dynamic parameter values.

```
<Request>
<Identifier>WTIMER</Identifier>
<StartDate>2014-02-02</StartDate>
</Request>
```

The following figure shows the message flow that SPSSJobIntegration uses.

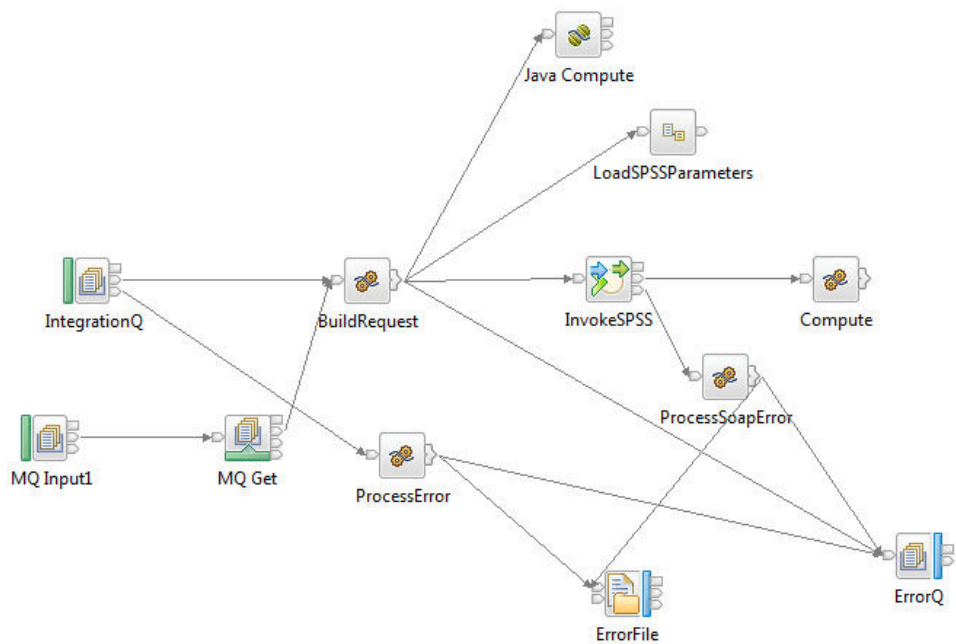


Figure 3. SPSSJobIntegration message flow

The following figure shows the parameters that the web service configuration requires.

webservice	
url	http://9.122.121.208:9080/process/services/ProcessManagement
jobLocationURI	spsscr:///id=569103e53065d83300000144f8d1202dbe9e
parameters	
parameter	
name	RunDateInFormatYYYYMMDDHyphenSeparated
value	StartDate
type	dynamic
parameter	
name	ServiceTablQtyMultiplie
value	1
type	static
parameter	
name	IsRunDateEqServerDate
value	0
type	static
notificationEnabled	true

Figure 4. Required parameters for the web service configuration

To trigger SPSSJobIntegration, a data preparation flow or Timer flow node that is specific to the use case must be created. The remaining parameters and fields that are required to invoke the SPSS batch job are taken from the orchestration XML file.

If there are only static parameters that are required for an SPSS job invocation, AutoTrigger and SPSSJobIntegration flows can be used together as a combination. In scenarios where dynamic parameters are involved, an extra data preparation flow that prepares the data, as required by the use case, is also needed.

If a new feature orchestration is introduced in the orchestration XML file, the PMQBatchIntegration flow must be restarted.

Tip: If you want to change configurable properties in the XML file, edit the changes in a local copy of the file, and replace the run time XML file with the modified file to avoid problems that are due to file locking.

The following table describes the features that use batch orchestration to run a scheduler flow, an SPSS job, an external flow, or a combination of flows and jobs, based on the scheduler and web service configuration in the batch orchestration XML file. Depending on your needs, configurations for flows and jobs can be updated at run time in the XML file.

Table 4. Features that use batch orchestration

Feature name	Purpose
Parametric	Triggers the Parametric adapter invocation at the scheduled time once a day, and modifies the default Parametric SubUseCase name.
Inspection	Triggers the Inspection adapter invocation at the scheduled time once a day.
Warranty	Triggers the Warranty SPSS job at the scheduled time once a day through the web service.
HS training	Triggers the HS Training SPSS job at the scheduled time once every 90 days through the web service.
FBA training	Triggers the Feature-based Analytics training SPSS job at the scheduled time once every 90 days through the web service.
IHS training	Triggers the Integrated Health Score training SPSS job at the scheduled time once every 90 days through the web service.
TopNFailure	Triggers the TopNFailure feature at the scheduled time once a day, and invokes the SPSS job for TopNFailure Analytics and Event generation through the web service.
Maintenance	Triggers the Maintenance feature at the scheduled time once a day, and invokes the SPSS job for Maintenance Analytics and Event generation through the web service.
Distribution transformer current aging	Triggers the Distribution Transformer Current Aging SPSS job at the scheduled time once a day through the web service.
Distribution transformer projected aging	Triggers the Distribution Transformer Projected Aging SPSS job at the scheduled time once every 180 days through the web service.
Pole FBA	Triggers the Pole FBA SPSS job at the scheduled time once every 30 days through the web service.

Chapter 4. Master data

Master data is the type of resource that you want to manage, for example, definition of a material or production process.

Master data can come from manufacturing engineering systems (MES) such as IBM Maximo, or other existing data sources. IBM InfoSphere® Master Data Management Collaboration Server can be used to complete gaps in the data from these sources or consolidate data from multiple sources. You can also add attributes, create relationships between items, or define data that you do not have another source for. For example, add hierarchy information to indicate which pieces of equipment belong to which site, in which location or classify resources into groups. In a report, the hierarchies and groups can be displayed as additional information or used as drill downs and filters.

Master data is normally loaded by one of the supplied connectors or the Flat File API. The connectors and the Flat File API use IBM Integration Bus flows to transform the data into the form that is required and update the data in the IBM Predictive Maintenance and Quality database.

Master data process

When a file is placed in the file input directory, IBM Integration Bus reads and processes it and then removes it from the directory. IBM Integration Bus stores and retrieves data from the database as required.

The response file indicates whether the operation was successful, and lists any results. If errors occur, a log file is written to the error directory.

The following diagram shows the flow of a file request and its response.

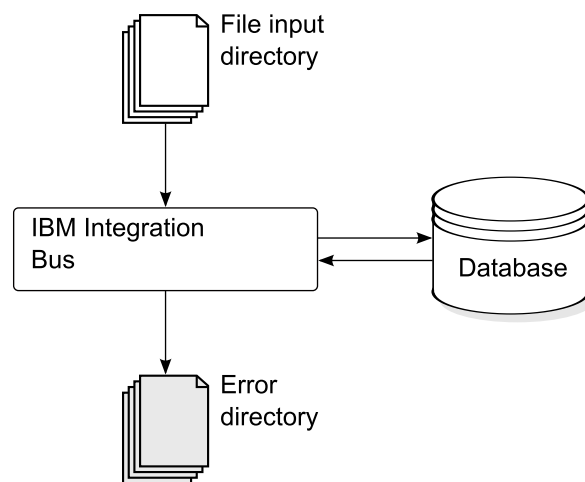


Figure 5. Master data process

Data organization

IBM Predictive Maintenance and Quality processes the following kinds of data:

- Master data supplies IBM Predictive Maintenance and Quality with information about the context in which events occur. Master data includes descriptions of the devices that produce events, the location where events occur, and the material that is used in an event.
- Metadata defines how IBM Predictive Maintenance and Quality processes received events. For more information, see “Metadata in the API” on page 189.
- Event data supplies IBM Predictive Maintenance and Quality with information that you want to measure about an event. For more information, see “How events are processed” on page 51.

The flat file application programming (API) interface

IBM Predictive Maintenance and Quality master data is supplied, accessed, modified, or removed using the flat file API. For more information, see Appendix B, “The flat file API,” on page 175.

File format and location

Master data and event data must be in a format that IBM Predictive Maintenance and Quality can recognize. The default file format is flat file, comma separated (.csv) format. Other file formats can be used, but you must create extra IBM Integration Bus flows.

File location

The file location is determined by the MQSI_FILENODES_ROOT_DIRECTORY environment variable. The file location is configured during the installation process.

This location contains the following sub folders:

- \masterdatain
used for loading master data and metadata files
- \eventdatain
used for loading event data files
- \error
used to report errors that occur while loading data
- \maximointegration
used for loading data files from IBM Maximo
- \control
- \restricted
- \properties

File names

Files must follow this naming convention:

record_name_operation.csv*

For example, a file that contains a set of resource records to be added to IBM Predictive Maintenance and Quality might be named:

resource_upsert_01.csv

File format

The .csv file format is used by default:

- Each line in a file is a record, and contains a sequence of comma-separated values. If a value contains a comma, the value must be contained within double quotation marks ",".
- Each record normally includes a code value (or combination of values) that uniquely identifies the record. These code values are sometimes known as business keys. Because this code value is a unique identifier for a row, it is used in other files as a way to reference that particular row. For example, in a file that contains a list of resources, the row for a resource can contain a location value. The location value is the code that is used to identify a location record.
- Sometimes a code value is required but is not applicable for a particular record. In this case, the special code **-NA-** must be used. For example, to avoid defining a location for a particular resource, use the code **-NA-** for the location value. The code value cannot be changed.
- In addition to a code value, a record typically has a name value. Both code and name value can hold the same value. However, while the code value must be unique for each row and is not normally shown to users, the name is visible in reports and dashboards. The name can be changed, unlike the code value.

The following example shows the format for a `location.csv` file. The command must be on a single line, not as shown here:

```
location_cd,location_name,region_cd,region_name,country_cd,country_name,
state_province_cd,state_province_name,city_name,latitude,longitude,
language_cd,tenant_cd,is_active
RAVENSWOOD,Ravenswood,NORTH AMERICA,North America,USA,United States,
CA,California,Los Angeles,34.0522,118.2428,,
TARRAGONA,Tarragona,EUROPE,Europe,UK,United Kingdom,ENGLAND,England,
London,51.5171,0.1062,,1
```

The following example shows codes that are used to identify records and used to reference other records. The codes that are used to identify a resource record are different from other records because a resource record is identified by both `Resource_CD1` and `Resource_CD2`, or by `operator_cd`.

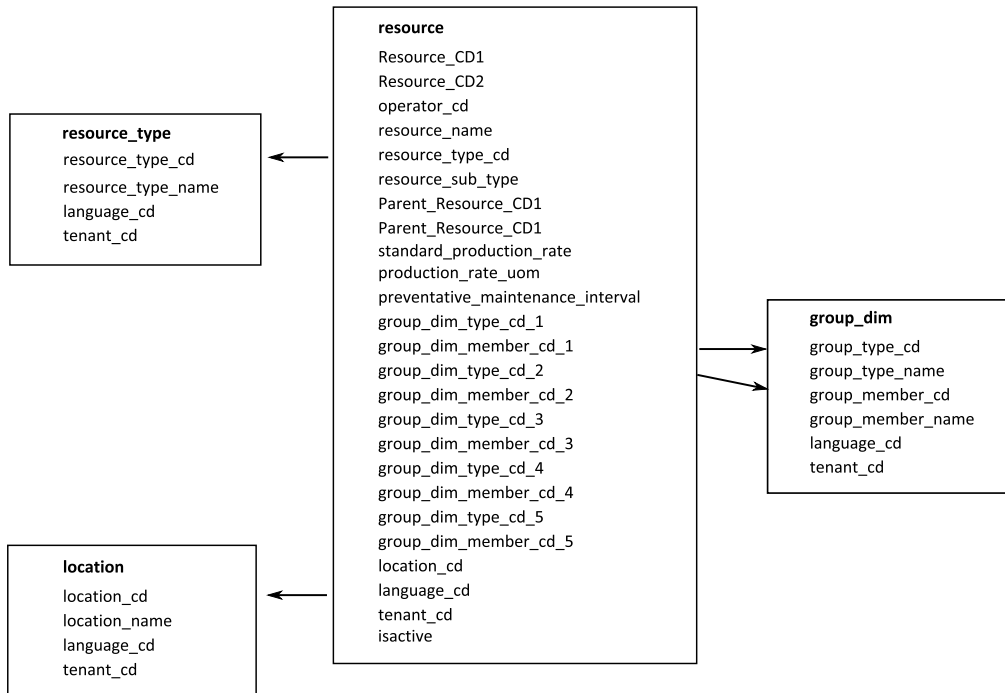


Figure 6. Codes used to identify and to reference records

Modifying a resource or process parent

If you must change a resource or process parent, you must reload the resource or process and all of its children. Modify the parent in a master data .csv file that contains all these rows, and resubmit the file.

Security

Implement security by restricting access to the directories used to supply files for the API.

Master data using InfoSphere MDM Collaboration Server

You can use IBM InfoSphere Master Data Management Collaboration Server to complete gaps in the data from external sources or consolidate data from multiple sources. You can also add attributes, create relationships between items, or define data that you do not have another source for.

For example, add hierarchy information to indicate which pieces of equipment belong to which site, in which location or classify resources into groups. In a report, the hierarchies and groups can be displayed as additional information or used as drill-downs and filters.

IBM InfoSphere Master Data Management Collaboration Server is model driven: you create a specification, and then define the fields. It automatically generates the user interface for the fields, for example, lookup tables, and date pickers. You can embed images in the data such as a picture of an asset.

A model for InfoSphere MDM Collaboration Server is provided with IBM Predictive Maintenance and Quality that simplifies the configuration. To use this model, you must do the following configuration steps.

1. Set the environment variable *PMQ_HOME* to the root of the IBM Predictive Maintenance and Quality installation directory.
2. Create a company for IBM Predictive Maintenance and Quality, see “Creating a company in IBM InfoSphere MDM Collaboration Server” on page 28.
3. Import the metadata (Company deployment), see “Importing metadata into InfoSphere MDM Collaboration Server” on page 31.
4. Configure the InfoSphere MDM Collaboration Server user interface, see “Configuring the IBM InfoSphere MDM Collaboration Server user interface” on page 29.

There are some specific guidelines that you must follow to ensure that you get the results that you expect. See “Guidelines for managing data in IBM InfoSphere MDM Collaboration Server” on page 29.

For additional information about using InfoSphere MDM Collaboration Server, see *Collaborative authoring with InfoSphere MDM Collaboration Server*. This is available from IBM Master Data Management Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSWSR9_11.0.0).

IBM Master Data Management Collaboration Server dynamic references

IBM Master Data Management Collaboration Server tasks use several dynamic references.

The following table describes the variables that are used in the InfoSphere MDM Collaboration Server tasks.

Table 5. Dynamic references

Reference	Description
<i>\$PMQ_HOME</i>	IBM Predictive Maintenance and Quality installation home directory.
<i>mdm_install_dir</i>	The root directory of the InfoSphere MDM Collaboration Server installation. \$TOP is an environment variable that is configured with InfoSphere MDM Collaboration Server by default, which points to this location.
<i>mdm_server_ip</i>	The IP address of InfoSphere MDM Collaboration Server, as seen by other IBM Predictive Maintenance and Quality servers, such as IBM Integration Bus.
<i>pmq_mdm_content_zip</i>	The full path to the content compressed file on the server file system.
<i>mdm_data_export_dir</i>	The directory, mount point, or symbolic link on the InfoSphere MDM Collaboration Server where data exports are configured to be written. The default is <code><\$PMQ_HOME>/data/export/mdm</code> .
<i>wmb_server_ip</i>	The IP address of IBM Integration Bus server, as seen by other IBM Predictive Maintenance and Quality servers.
<i>wmb_fileapi_input_dir</i>	The directory where input data files are to be placed for loading into the IBM Predictive Maintenance and Quality database. The directory can be local or remote. The file location is determined by the MQSI_FILENODES_ROOT_DIRECTORY environment variable. The file location is configured during the installation process.
<i>company_code</i>	The company code for InfoSphere MDM Collaboration Server. Make the code short and easy to remember because it must be entered during each login, for example, IBMPMQ.

Table 5. Dynamic references (continued)

Reference	Description
<i>company_name</i>	The display name of the company in InfoSphere MDM Collaboration Server, for example, IBMPMQ.

Creating a company in IBM InfoSphere MDM Collaboration Server

You must create a company before you can import IBM Predictive Maintenance and Quality metadata into IBM InfoSphere Master Data Management Collaboration Server. A company is similar to the concept of a project.

About this task

For information about the variables used, see “IBM Master Data Management Collaboration Server dynamic references” on page 27.

Procedure

1. Stop the InfoSphere MDM Collaboration Server service.
 - a. Change the directory to `cd <mdm_install_dir>/bin/go` where `<mdm_install_dir>` is the root directory of the InfoSphere MDM Collaboration Server installation.
 - b. Run the **stop_local.sh** command: `./stop_local.sh`
2. Run the company creation script.
 - a. Change the directory to `cd <mdm_install_dir>/bin/db`
 - b. Run the **create_cmp.sh** command: `./create_cmp.sh -code=<company_code> --name=<company_name>`
3. Start the InfoSphere MDM Collaboration Server service.
 - a. Change the directory to `cd <mdm_install_dir>/bin/go`
 - b. Run the **start_local.sh** command: `./start_local.sh`
4. Log in and verify the company. Open your web browser and enter the URL for the InfoSphere MDM Collaboration Server web server, for example: `http://<mdm_host_name>:7507/utills/enterLogin.jsp`
 The following default users are created for the new company:

Table 6. Default roles, users, and passwords created for a new company

Role	User name	Password
Administrator	Admin	trinitron
Basic User	Basic	trinitron

5. Change the default passwords for both the administrator, and for the basic user. You do this in the **Data Model Manager** module > **User Console**.

What to do next

The next step is to import the IBM Predictive Maintenance and Quality metadata into the InfoSphere MDM Collaboration Server.

Configuring the IBM InfoSphere MDM Collaboration Server user interface

Add the IBM Predictive Maintenance and Quality objects in the IBM Master Data Management Collaboration Server navigation area to make it easier to manage data.

Procedure

1. In InfoSphere MDM Collaboration Server, click **Please select a module to add**. A drop-down list is displayed.
2. Select all of the following modules from the **Catalog** module type.
 - **Asset**
 - **Locations**
 - **Material Types**
 - **Processes**
 - **Products**
 - **Suppliers**
3. Select **Groups by Type** from the **Hierarchy** module type.

What to do next

You can customize group types to suit the needs of the project.

1. In the **Groups by Type** hierarchy, choose a group type, and customize it as required with a new code or name.
2. Save the changes.
3. Update the **Group Hierarchy Lookup** by clicking **Product Manager > Lookup Tables, Lookup Table Console**.
4. Update the group type record with the new group type code.

Guidelines for managing data in IBM InfoSphere MDM Collaboration Server

You must follow these guidelines to manage data in IBM InfoSphere Master Data Management Collaboration Server to ensure that you get the results that you expect.

Assets

Define assets in the **Unassigned** category.

You can use the default hierarchy to organize items, but the hierarchy is not used by IBM Predictive Maintenance and Quality.

Group assignments:

- Can be assigned up to five groups from the **Groups by Type** hierarchy.
- Each assignment must be from a different group type.
- Must be assigned to Group (Level 2), not Group Type (Level 1.)

Groups

Groups are managed by using the group hierarchy rather than a catalog. Only categories are defined, not items.

The first level must be group type.

The second level must be groups.

Locations

Define the locations as follows:

- The first level must be **Region** (Location Type=Region).
- The second level must be **Country** (Location Type=Country).
- The third level must be **State** (Location Type=State / Province).

The location items must be defined only under State / Province (only on a leaf node).

Material types, processes, products, and suppliers

Define items in the **Unassigned** category.

You can use the default hierarchy to organize items, but the hierarchy is not used by IBM Predictive Maintenance and Quality.

Configuring and running data exports

To integrate IBM InfoSphere Master Data Management Collaboration Server into IBM Predictive Maintenance and Quality, data export files must be sent to the data input directory for the flat file API on the IBM Integration Bus server.

Before you begin

For information about the variables that are used, see “IBM Master Data Management Collaboration Server dynamic references” on page 27.

About this task

The IBM Integration Bus file location is determined by the `MQSI_FILENODES_ROOT_DIRECTORY` environment variable, and the folder is named `\masterdatain`. The file location is configured during the installation process.

Procedure

1. On the IBM Integration Bus server, ensure that the Network File System (NFS) is configured to run with the following command.

```
/sbin/chkconfig nfs on
```

2. Share the data input directory for the flat file API by adding the following line to `/etc/exports`. Create the directory if it does not exist.

```
<wmb_fileapi_input_dir> <mdm_server_ip>(rw)
```

3. Ensure that sufficient permissions are set on the data input directory.

The following example grants read and write permissions to all users and groups. If you require a more secure configuration, ensure that users, groups, and permissions are consistent with those on the InfoSphere MDM Collaboration Server so that NFS operates correctly.

```
chmod 777 <wmb_fileapi_input_dir>
```

4. Restart the NFS service for the settings to take effect.

```
service nfs restart
```


5. On the InfoSphere MDM Collaboration Server, ensure that the data export directory exists. If it does not, create the directory.

```
mkdir <mdm_data_export_dir>
```

6. Mount the remote flat file API input directory with NFS.

```
mount -t nfs -o rw wmb_server_ip:wmb_fileapi_input_dir mdm_data_export_dir
```

7. Test NFS sharing.

- a. Create a test file on the InfoSphere MDM Collaboration Server.

```
echo <"NFS Test File"> <mdm_data_export_dir>/nfstest.txt
```

- b. Check for the test file on the IBM Integration Bus server:

```
cat <wmb_fileapi_input_dir>/nfstest.txt
```

Results

If the file content is displayed, NFS is working. If you have problems, search for “Red Hat Linux NFS documentation” online for detailed information.

What to do next

To run a data export, in the InfoSphere MDM Collaboration Server Reports Console, select the export and click the **Run** icon. Data export files are written to `$PMQ_HOME/<mdm_data_export_dir>`. The default is `$PMQ_HOME/data/export/mdm`.

Importing metadata into InfoSphere MDM Collaboration Server

You must import IBM Predictive Maintenance and Quality data into IBM Master Data Management Collaboration Server before you can use MDM to manage data.

About this task

For information about the variables that are used, see “IBM Master Data Management Collaboration Server dynamic references” on page 27.

Procedure

Use the following command to import data into InfoSphere MDM Collaboration Server. The command must be on a single line, not as shown here.

```
<mdmce_install_dir>/bin/importCompanyFromZip.sh  
--company_code=<company_code>  
--zipfile_path=IBMPMQ.zip
```

Example

See the following example.

```
$TOP/bin/importCompanyFromZip.sh --company_code=IBMPMQ --zipfile_path  
=$PMQ_HOME/content/IBMPMQ.zip
```

`$TOP` is a built-in IBM InfoSphere Master Data Management Collaboration Server environment variable, which points to the root Master Data Management Collaboration Server directory.

Solution XML file

The solution XML file defines the master data. The master tables and supporting tables are defined so that database tables may be generated and upserts carried out.

The solution XML file defines the following types of tables:

- Master tables
- Event tables
- Profile or KPI tables

The LANGUAGE table and the columns are defined as shown in the following XML code:

```
<table table_cd="LANGUAGE" is_surrogate_primary_key="true"
      validator_class="com.ibm.pmq.master.validators.LanguageValidate">
  <column column_cd="LANGUAGE_CD" type="string" size="50" is_key="true"/>
  <column column_cd="LANGUAGE_NAME" type="string" size="200"/>
  <column column_cd="DEFAULT_IND" type="int"/>
</table>
```

The TENANT table and the columns are defined as shown in the following XML code:

```
<table table_cd="TENANT" is_surrogate_primary_key="true"
      validator_class="com.ibm.pmq.master.validators.TenantValidate">
  <column column_cd="TENANT_CD" type="string" size="100" is_key="true"/>
  <column column_cd="TENANT_NAME" type="string" size="200"/>
  <column column_cd="DEFAULT_IND" type="int"/>
</table>
```

The definitions of the LANGUAGE, TENANT, CALENDAR, EVENT_TIME, and KEYLOOKUP tables must not be modified and must be included in the solution XML file.

Master tables include language and tenant support. They are defined by using attributes of the table. For example, the following definition of the Master_Location table includes the attributes is_multilanguage, is_multitenant, and is_row_deactivateable. The value of "true" indicates that the table is multi-language, multi-tenant, and the table includes a column that indicates whether the row is enabled (active) or disabled (deactivated):

```
<table table_cd="MASTER_LOCATION"
      is_multilanguage="true" is_multitenant="true" is_row_deactivateable="true"
      is_surrogate_primary_key="true"
      validator_class="com.ibm.pmq.master.validators.LocationValidate">
  <column column_cd="LOCATION_CD" is_key="true" size="100"
type="string"/>
  <column column_cd="LOCATION_NAME" is_key="false" size="1024"
type="string"/>
  <column column_cd="REGION_CD" is_key="false" size="50"
type="string" is_nullable="true"/>
  <column column_cd="REGION_NAME" is_key="false" size="200"
type="string" is_nullable="true"/>
  <column column_cd="COUNTRY_CD" is_key="false" size="50"
type="string" is_nullable="true"/>
  <column column_cd="COUNTRY_NAME" is_key="false" size="200"
type="string" is_nullable="true"/>
  <column column_cd="STATE_PROVINCE_CD" is_key="false" size="50"
type="string" is_nullable="true"/>
  <column column_cd="STATE_PROVINCE_NAME" is_key="false" size="200"
type="string" is_nullable="true"/>
  <column column_cd="CITY_NAME" is_key="false" size="200"
type="string" is_nullable="true"/>
```

```

type="string" is_nullable="true"/>
  <column column_cd="LATITUDE" is_key="false" size="10,5"
type="decimal" is_nullable="true"/>
  <column column_cd="LONGITUDE" is_key="false" size="10,5"
type="decimal" is_nullable="true"/>
</table>

```

References

The tables defined in the solution XML file (event, master data, and profile) may also define references to master data tables. For example, Master_Product_Parameters references the Master_Product table. To reference a specific Master_Product row, the flows for Master_Product_Parameters take the business keys Product_Cd and Product_Type_Cd as input parameters in the CSV file. The following definition for Master_Product_Parameters is an example of how to define a reference. Product_Id is an identifier of the reference to the Master_Product table. The business keys of the Master_Product table, Product_type_cd, and Product_cd, along with Tenant_cd, are used to reference a Master_Product row:

```

<table table_cd="MASTER_PRODUCT_PARAMETERS"
  is_multilanguage="true" is_multitenant="true">
  <column column_cd="PARAMETER_NAME" type="string" size="50"
  is_key="true"/>
  <column column_cd="PARAMETER_VALUE" type="double"
  is_key="false"/>
  <reference reference_cd="PRODUCT_ID"
  table_reference="MASTER_PRODUCT" is_key="true"/>
</table>

```

The following example shows a more explicit table definition for Master_Product_Parameters. This method can be used to make the column names different than the business keys. That is, when table_column_cd is different from reference_column_cd. You must use this mapping to have unique reference_column_cd values when there is more than one reference to the same table:

```

<table table_cd="MASTER_PRODUCT_PARAMETERS"
  is_multilanguage="true" is_multitenant="true">
  <column column_cd="PARAMETER_NAME" type="string" size="50"
  is_key="true"/>
  <column column_cd="PARAMETER_VALUE" type="double"
  is_key="false"/>
  <reference reference_cd="PRODUCT_ID"
  table_reference="MASTER_PRODUCT" is_key="true">
  <column_mapping table_column_cd="PRODUCT_CD" reference_column_cd="PRODUCT_CD"/>
  <column_mapping table_column_cd="PRODUCT_TYPE_CD"
  reference_column_cd="PRODUCT_TYPE_CD"/>
  </reference>
</table>

```

Hierarchy table structures

The solution XML file manages the hierarchical structures that are used in IBM Predictive Maintenance and Quality. IBM Predictive Maintenance and Quality maintains hierarchical structures for two Master tables, Resource and Process.

Master_Resource_hierarchy is generated based on the solution XML. The following example shows the definition of Master_Resource in the solution XML file. The self_reference element means that there is a circular reference to the table. The circular reference is required to maintain the hierarchy. The number_of_levels

property defines the number of levels of hierarchy. The duplicate_column_cd element refers to the column names that appear across each level of the defined number_of_levels property:

```
<self_reference reference_cd="PARENT_RESOURCE_ID" number_of_levels="10">
  <column_mapping table_column_cd="RESOURCE_CD1"
reference_column_cd="PARENT_RESOURCE_CD1" />
  <column_mapping table_column_cd="RESOURCE_CD2"
reference_column_cd="PARENT_RESOURCE_CD2" />
  <duplicate_column_cd>RESOURCE_CD1</duplicate_column_cd>
  <duplicate_column_cd>RESOURCE_CD2</duplicate_column_cd>
  <duplicate_column_cd>RESOURCE_NAME</duplicate_column_cd>
</self_reference>
```

Master_Process_Hierarchy is generated based on the solution XML. The following example shows the definition of Master_Process in the solution XML file. For Master_Process_Hierarchy, hierarchical information for Process_CD and Process_Name is maintained across five levels:

```
<self_reference
reference_cd="PARENT_PROCESS_ID" number_of_levels="5">
  <column_mapping table_column_cd="PROCESS_CD"
reference_column_cd="PARENT_PROCESS_CD"/>
  <duplicate_column_cd>PROCESS_CD</duplicate_column_cd>
  <duplicate_column_cd>PROCESS_NAME</duplicate_column_cd>
</self_reference>
```

IBM Maximo Asset Management

Master data and event data can be supplied from IBM Maximo to IBM Predictive Maintenance and Quality. Recommended actions that are generated by IBM Predictive Maintenance and Quality can also be passed to IBM Maximo Asset Management.

IBM Maximo Asset Management is not installed as part of IBM Predictive Maintenance and Quality. If required, it must be purchased separately. However, IBM Predictive Maintenance and Quality contains adapters for IBM Maximo, which allow data integration.

How master data is mapped in IBM Maximo Asset Management

As an example, the following tables in IBM Predictive Maintenance and Quality can be populated from the default Maximo object model.

group_dim table

The records in the group_dim table provide classifications for resources. You can have up to five classifications for each resource. The classifications might vary.

Table 7. Fields in the group_dim table

Field	Type	Required or optional	Maximo objects/attributes
group_type_cd	string(50)	Required	"MXCLASSIFICATION"
group_type_name	string(200)	Required	"Maximo Classification"
group_member_cd	string(50)	Required	CLASSTRUCTURE.CLASSSTRUCTUREID
group_member_name	string(200)	Required	CLASSTRUCTURE.DESCRPTION

Location table

The location table contains the location of a resource or event, such as a room in a factory or a mine site. In Maximo, this information is stored as a LOCATIONS object and in its associated SERVICEADDRESS object.

Table 8. Fields in the location table

Field	Type	Required or Optional	Maximo objects/attributes
location_cd	string(50)	Required	SERVICEADDRESS.ADDRESSCODE
location_name	string(200)	Required	SERVICEADDRESS.DESCRPTION
region_cd	string(50)	Optional, region_cd and region_name must be supplied together	SERVICEADDRESS.REGIONDISTRICT
region_name	string(200)	Optional	SERVICEADDRESS.REGIONDISTRICT
country_cd	string(50)	Optional, country_cd and country_name must be supplied together	SERVICEADDRESS.COUNTRY
country_name	string(200)	Optional	SERVICEADDRESS.COUNTRY
state_province_cd	string(50)	Optional, country_cd and country_name must be supplied together	SERVICEADDRESS.STATEPROVINCE
state_province_name	string(200)	Optional	SERVICEADDRESS.STATEPROVINCE
city_name	string(200)	Optional	SERVICEADDRESS.CITY
latitude	float (in decimal degrees)	Optional	SERVICEADDRESS.LATITUDE
longitude	float (in decimal degrees)	Optional	SERVICEADDRESS.LONGITUDE

resource table

A resource defines resources of type asset or agent. An asset is a piece of equipment. An agent is the operator of the equipment. Some asset resources might form a hierarchy (for example, a truck is a parent of a tire). Asset information imported from Maximo includes the asset type, classification, and location.

Table 9. Fields in the resource table

Field	Type	Required or Optional	Maximo objects and attributes
Resource_CD1	string(50)	Either serial_no and model or operator_cd are required	ASSET.ASSETNUM
Resource_CD2	string(50)		ASSET.SITEID
resource_name	string(500)	Required	ASSET.DESCRPTION
resource_type_cd	string(50)	Required	

Table 9. Fields in the resource table (continued)

Field	Type	Required or Optional	Maximo objects and attributes
resource_sub_type	string(50)	Optional	ASSET.ASSETTYPE
parent_resource_serial_no	string(50)	Optional (parent_resource_serial_no and parent_resource_model should be supplied together)	ASSET.PARENT
parent_resource_model	string(50)	Optional	ASSET.SITEID
parent_resource_operator_cd	string(50)	Optional	
standard_production_rate	float	Optional	
production_rate_uom	string(40)	Optional	
preventative_maintenance_interval	float	Optional	
group_dim_type_cd_1	string(50)	Group codes are required but a NA value can be specified for a corresponding type and a member	"MXCLASSIFICATION"
group_dim_member_cd_1	string(50)		ASSET.CLASSSTRUCTUREID
group_dim_type_cd_2	string(50)		
group_dim_member_cd_2	string(50)		
group_dim_type_cd_3	string(50)		
group_dim_member_cd_3	string(50)		
group_dim_type_cd_4	string(50)		
group_dim_member_cd_4	string(50)		
group_dim_type_cd_5	string(50)		
group_dim_member_cd_5	string(50)		
location_cd	string(50)	Required but a NA code can be specified	ASSET.SADDRESSCODE

Mapping master data in IBM Maximo Asset Management

IBM Predictive Maintenance and Quality includes sample flows that import assets, classifications, and ServiceAddress objects from the default Maximo object model. To enable these flows, master data must be exported out of IBM Maximo as XML files, and is later placed into the \maximointegration folder.

About this task

Asset data that is managed in IBM Maximo is mirrored in IBM Predictive Maintenance and Quality. When data is modified in IBM Maximo, it is automatically updated in IBM Predictive Maintenance and Quality. Data that comes from IBM Maximo must be updated and maintained in IBM Maximo. It is not possible for changes that are made in IBM Predictive Maintenance and Quality to be propagated back to IBM Maximo.

A Maximo Publish Channel is used to export assets, classifications, and the **ServiceAddress** attribute. You must invoke the channel manually initially to populate the IBM Predictive Maintenance and Quality database. After, the channel is automatically triggered whenever the contents of one of these objects changes.

For more information, see IBM Maximo Asset Management Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSWK4A>).

Procedure

1. Create an object structure in IBM Maximo based on the base object structures available in IBM Maximo Asset Management.

IBM Predictive Maintenance and Quality supports data mapping for three object structures: SPASSET, SPSERVICEADDRESS, and SPCLASSIFICATION.

These object structures are inherited from base object structures in IBM Maximo: ASSET, SERVICEADDRESS, and CLASSSTRUCTURE.

When the object structure is created, use the **Exclude/Include fields** option from the **Select Action** menu to include or exclude fields.

For more information, see *Object structures* in the IBM Maximo Asset Management, *Integrating data with external applications, Integration Components* online documentation.

2. Create the following publish channels:

- SPCLASSIFICATIONCHANNEL_R with object structure SPCLASSIFICATION
- SPPUBLISHCHANNEL_R with object structure SPASSET
- SPSAPUBLISHCHANNEL with object structure SPSERVICEADDRESS

For each publish channel, perform the following action:

- Configure the endpoint to be XML.

For more information, see *Publish channels* in the IBM Maximo Asset Management, *Integrating data with external applications, Integration Components, Channels and services* online documentation.

3. Create an external system and configure the corresponding endpoint for the external system as XML.

The name of the external system must be SPEXTSYSTEM.

Configure the location as the \maximointegration folder. The location of the folder is determined by the MQSI_FILENODES_ROOT_DIRECTORY environment variable.

When IBM Maximo and IBM Integration Bus are installed on different systems, this folder must be shared, or the exported files must be transferred to this folder.

4. Set up publish channels for the external systems.

- a. Name the publish channels as shown:

SPPUBLISHCHANNEL

For Asset.

SPCLASSIFICATIONCHANNEL

For Classification.

SPSAPUBLISHCHANNEL

For ServiceAddress.

- b. Select each publish channel in turn and click **Data Export** to export data.

The export screen supports a filter expression to export a subset of data. For example, if you want to export assets with a specific classification then you must enter a filter expression such as CLASSSTRUCTUREID='1012'.

To find the CLASSSTRUCTUREID that an asset belongs to, go to the **Specifications** tab of ASSET.

The **Specifications** tab contains classification information. The classification has a CLASSSTRUCTUREID associated with it, which you can see when you export the classification.

The exported XML is stored in the \maximo\integration folder.

5. Export the Object Structure schema:
 - a. Search and select the Object Structure for which the schema file must be generated.
 - b. Select **Generate Schema/View XML** action for that object structure. You can select the operation for which schema must be generated. Select the **Publish** operation.

The generated schema is stored in the same location as the data export XML files. These schema files correspond to the SPASSETService.xsd, SPCLASSIFICATIONService.xsd, and SPSERVICEADDRESSService.xsd files in the PMQMaximoIntegration IBM Integration Bus library.

Enabling master data loading in realtime mode

You can enable master data to load in realtime mode by creating publish channels and configuring their endpoints.

Procedure

1. Create new publish channel for real time master data loading.
 - a. Select **Integration > Publish Channels > New**.
 - b. Create the following publish channels:
 - SPCLASSIFICATIONCHANNEL_R, with object structure SPCLASSIFICATION
 - SPPUBLISHCHANNEL_R, with object structure SPASSET
 - SPSAPUBLISHCHANNEL, with object structure SPSERVICEADDRESS
 - c. For each publish channel, select **Action > Enable Event Listeners**, and then select the **Enable Listener** check box.
2. Configure the Web service endpoints.
 - a. Select **GoTo > Integration > Endpoint**.
 - b. Select **New Endpoint** and enter the following information:
 - In the **Endpoint Name** field, type AENDPOINT
 - In the **Handler type** field, type WEBSERVICE
 - In the **EndPointURL** field, type http://ESB_Node_IP_address:7800/meaweb/services/asset
 - In the **ServiceName** field, type asset
 - c. Select **New Endpoint** and enter the following information:
 - In the **Endpoint Name** field, type CENDPOINT
 - In the **Handler type** field, type WEBSERVICE
 - In the **EndPointURL** field, type http://ESB_Node_IP_address:7800/meaweb/services/classification
 - In the **ServiceName** field, type classification
 - d. Select **New Endpoint** and enter the following information:
 - In the **Endpoint Name** field, type SAENDPOINT
 - In the **Handler type** field, type WEBSERVICE
 - In the **EndPointURL** field, type http://ESB_Node_IP_address:7800/meaweb/services/serviceaddress

- In the **ServiceName** field, type serviceaddress
3. Configure the external system to associate the publish channels and endpoints to the external system for webservice event notification of workorders.
 - a. Select **GoTo > Integration > External Systems > filter** for EXTSYS2
 - b. Select **Publish channels > Add New Row**.
 - Enter SPCLASSIFICATIONCHANNEL : CENDPOINT
 - Select the **Enabled** check box.
 - c. Select **Publish channels > Add New Row**.
 - Enter SPPUBLISHCHANNEL : AENDPOINT
 - Select the **Enabled** check box.
 - d. Select **Publish channels > Add New Row**.
 - Enter SPSAPUBLISHCHANNEL : SAENDPOINT
 - Select the **Enabled** check box.

Importing event data from IBM Maximo Asset Manager

IBM Predictive Maintenance and Quality can be customized to import IBM Maximo work orders as events to record activities such as inspections and repairs.

You must do the following tasks:

1. Create a publish channel in IBM Maximo to export the work orders.
Take care not to import work orders that are created by IBM Predictive Maintenance and Quality.
Modify the WorkorderCreation flow to set the EXTREFID field to PMQ. When you import the work order, do not import work orders that have the EXTREFID field that is set to PMQ.
For more information, see IBM Maximo Asset Management Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSWK4A>).
2. Create a flow in IBM Integration Bus to consume these work orders, map them to the standard event format, and place them on the event processing queue.
3. Create profile variables to determine how these events are processed into key performance indicators (KPIs) and profiles. For more information, see “Profile variables” on page 55
4. Modify the event processing flow to ensure that these events trigger scoring for an appropriate predictive model. For more information, see “Event processing” on page 62.

Creating a work order service in IBM Maximo Asset Management

To create a work order, an enterprise service must be created in IBM Maximo. The enterprise service defines a web service with a WSDL file. The work order creation service is called by an IBM Integration Bus flow in IBM Predictive Maintenance and Quality.

Before you begin

You must configure a web service in IBM Maximo Asset Management to create work orders in IBM Predictive Maintenance and Quality.

Configure IBM Maximo to expose a web service corresponding to the service defined in the **MaximoWorkOrder.wsdl** file in the **PMQMaximoIntegration** IBM Integration Bus application.

For more information about creating an enterprise service, see IBM Maximo Asset Management Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSWK4A>).

Procedure

Create a Web Service from the default work order enterprise service (MXWOInterface).

1. In IBM Maximo Asset Management, go to the **Web Services Library, Select Action, Create Web Service, Create Web Service from Enterprise Service**.
2. Select **EXTSYS1_MXWOInterface** and click **Create**.
3. Click the generated web service name (**EXTSYS1_MXWOInterface**) and **Select Action, Deploy to Product Web Service Container, Deploy Web Service** and click **OK**.
4. Turn on the functionality in IBM Predictive Maintenance and Quality to create work orders in IBM Maximo based on recommendations from the default predictive models. In the IBM WebSphere® MQ Explorer, set the **MaximoTRIGGER** user-defined property for the **PMQIntegration** flow to **TRUE**.
 - a. In the IBM WebSphere MQ Explorer, go to **Brokers > MB8Broker > PMQ1**. Right-click the **PMQIntegration** node, and click **Properties**.
 - b. Click **User Defined Properties**.
 - c. Set the **MaximoTRIGGER** value to **TRUE**.
5. Set the server name in the **Web Service URL** property of the **InvokeWorkOrder** node to the name of the IBM Maximo host. This node is in the sample **WorkorderCreation.msgflow** flow in the **PMQMaximoIntegration** application.
 - a. In the IBM WebSphere MQ Explorer, go to **Brokers > MB8Broker > PMQ1 > PMQMaximoIntegration > Flows**, and click **Workordercreations.msgflow**.
 - b. In the graphical display, right-click the **InvokeWorkOrder** node and select **Properties**.
 - c. In the **Web Services URL** field, enter the URL of the IBM Maximo host.

Configuring work orders in Maximo

In Maximo, you can configure Maximo for OutBound work orders using either an XML file in batch mode or using a web service in realtime mode.

You can also configure Maintenance work orders to be updated with recommendations in IBM Predictive Maintenance and Quality (PMQ).

Configuring Maximo for OutBound work orders that use a web service

You can configure Maximo for OutBound work orders that use a web service in real-time mode.

Procedure

1. Define the object structure.
 - a. Edit the base object structures available in IBM Maximo Asset Management (MXWO) to add the Service Address object reference to it.

Tip: This ensures that work order events that are generated from Maximo contain the field reference that is related to the service address.

- b. Select **GoTo > Integration > Object Structure** and search for MXWO.
- c. Click the new row and enter the following information:
 - In the **Object** field, type WOSERVICEADDRESS.
 - In the **Parent Object** field, type WORKORDER.
 - In the **Object Location Path** field, type WOSERVICEADDRESS.
 - In the **Relationship** field, type SERVICEADDRESS.
2. To export the Object Structure schema for MXWO, select **Action > Generate Schema/View XML**.

The generated schema MXWOService.xsd is stored in the same location as the data export XML files. This schema is used for configuring in the mapping node of IIB for work order to event transformation.
3. Enable the Publish channel event listener.
 - a. Select **Publish Channel** and then select **MXWOInterface**. The work order publish channel appears.
 - b. Select **Action > Enable Event Listeners**.

The **Enable Listener** check box is enabled. See the following figure.
4. Add a processing rule for the publish channel MXWOInterface.
 - a. Select **New Row**.
 - b. Enter the following values:
 - In the **Rule** column, type PMQ.
 - In the **Description** column, type PMQ Maintenance related Rule.
 - In the **Action** column, specify SKIP.
 - In the **Enabled** column, select the check box.
 - c. Select **Add/Modify Conditions**.
 - d. Select **New Row**.
 - e. Specify the following values:
 - In the **Field** field, type DESCRIPTION.
 - In the **Evaluation Type** field, type NOTEQUALS.
 - In the **Evaluation When** field, type ALWAYS.
 - In the **Value** field, type MAINTENANCE.

A condition is added to skip the MAINTENANCE work order.
 - f. Select **New Row**.
 - g. Specify the following values:
 - In the **Field** field, select DESCRIPTION.
 - In the **Evaluation Type** field, select NOTEQUALS.
 - In the **Evaluation When** field, select ALWAYS.
 - In the **Value** field, select BREAKDOWN.

A condition is added to skip the BREAKDOWN work order.
5. Activate the JMS cron task.
 - a. Select **GoTo > System Configuration > Platform Configuration > Cron Task Setup**.
 - b. Filter on **JMSQSEQCONSUMER**.
 - c. Select the **SEQQOUT** cron task instance name.
 - d. Click **Active** and save the record.

- The JMS cron task is activated.
6. Configure the web service endpoint.
 - a. Select **GoTo > Integration > Endpoint**.
 - b. Select **New Endpoint** and enter the following information:
 - In the **Endpoint Name** field, type MXWOENDPOINT.
 - In the **Handler type** field, type WEBSERVICE.
 - In the **EndPointURL** field, type `http://ESB_Node_IP_address:7800/meaweb/services/MXWOInterface`.
 - In the **ServiceName** field, type OutboundW0Service
 - .
 7. Configure the external system to associate publish channels and endpoints to the external system for web service event notification of work orders.
 - a. Select **GoTo > Integration > External Systems > New External System**
 - b. Enter the following information:
 - In the **System** field, type EXTSYS2.
 - In the **Description** field, type PMQ External System.
 - In the **EndPoint** field, type MXXMLFILE.
 - In the **Outbound Sequential Queue** field, type `jms/maximo/int/queues/sqout`.
 - In the **Inbound Sequential Queue** field, type `jms/maximo/int/queues/sqin`.
 - In the **Inbound Continuous Queue** field, type `jms/maximo/int/queues/cqin`.
 - Select the **Enabled** check box.
 - c. Select **Publish channels > Add New Row**.
 - Add a New Row to add MXWOInterface to the publish channel with Endpoint as MXWOENDPOINT.
 - Select the **Enabled** check box.

Configuring Maximo for OutBound work orders using an XML file

You can configure Maximo for OutBound work orders using an XML file in batch mode.

Procedure

1. Create a new publish channel SPWO.
 - a. Select **GoTo > Integration > Publish Channels**.
 - b. Enter the following information:
 - In the **Publish Channel** field, type SPWO.
 - In the **Description** field, type PMQ WorkOrder Publish Channel.
 - In the **Object Structure** field, type MXWO.
- See the following figure.

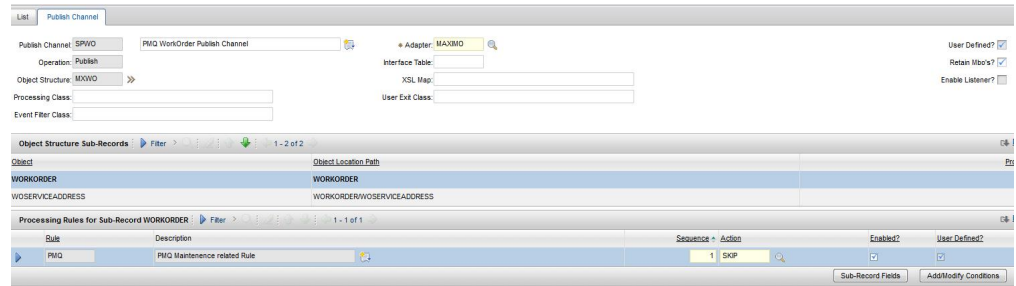


Figure 7. Create a new publish channel SPWO

2. Add a new processing rule for the publish channel SPWO.
 - a. Select **New Row**.
 - b. Specify the following values:
 - In the **Rule** column, type PMQ.
 - In the **Description** column, type PMQ Maintenance related Rule.
 - In the **Action** column, specify SKIP.
 - In the **Enabled** column, select the checkbox.
- See the following figure.

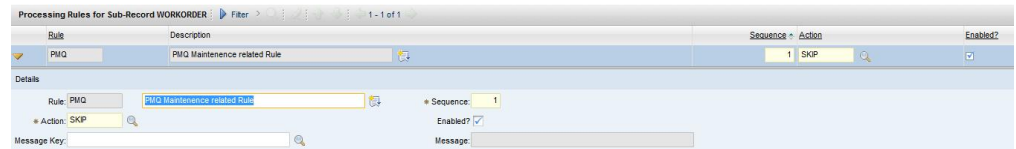


Figure 8. Adding a new processing rule for the publish channel SPWO

- c. Select **Add/Modify Conditions**.
- d. Select **New Row** under XML field evaluation.
- e. Specify the following values:
 - In the **Field** field, specify DESCRIPTION.
 - In the **Evaluation Type** field, specify NOTEQUALS.
 - In the **Evaluation When** field, specify ALWAYS.
 - In the **Value** field, specify MAINTENANCE.

A condition is added to skip the MAINTENANCE work order. See the following figure.

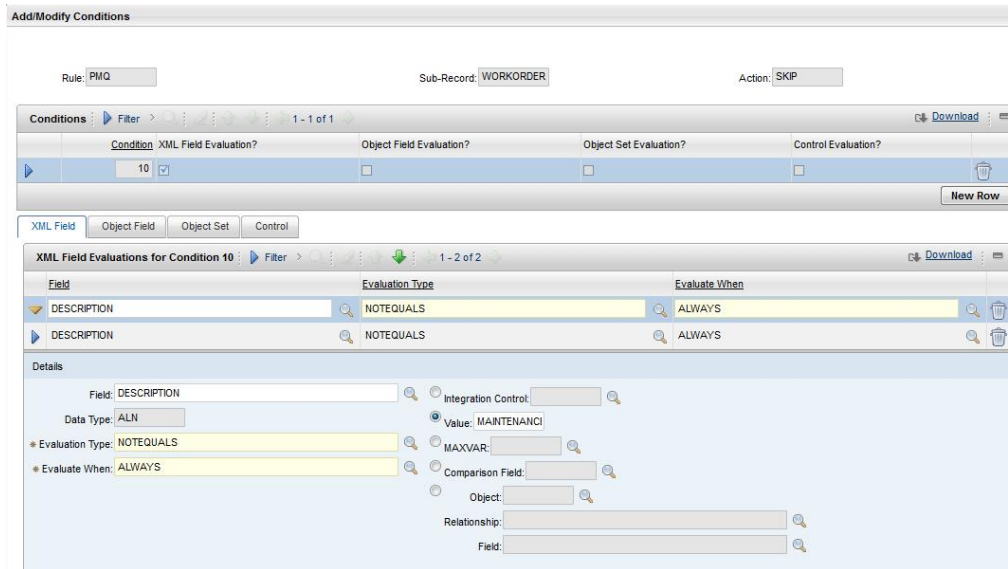


Figure 9. Adding a condition to skip the MAINTENANCE work order

- f. Select **New Row** under XML field evaluation.
- g. Specify the following values:
 - In the **Field** field, specify DESCRIPTION.
 - In the **Evaluation Type** field, specify NOTEQUALS.
 - In the **Evaluation When** field, specify ALWAYS.
 - In the **Value** field, specify BREAKDOWN.

A condition is added to skip the BREAKDOWN work order. See the following figure.

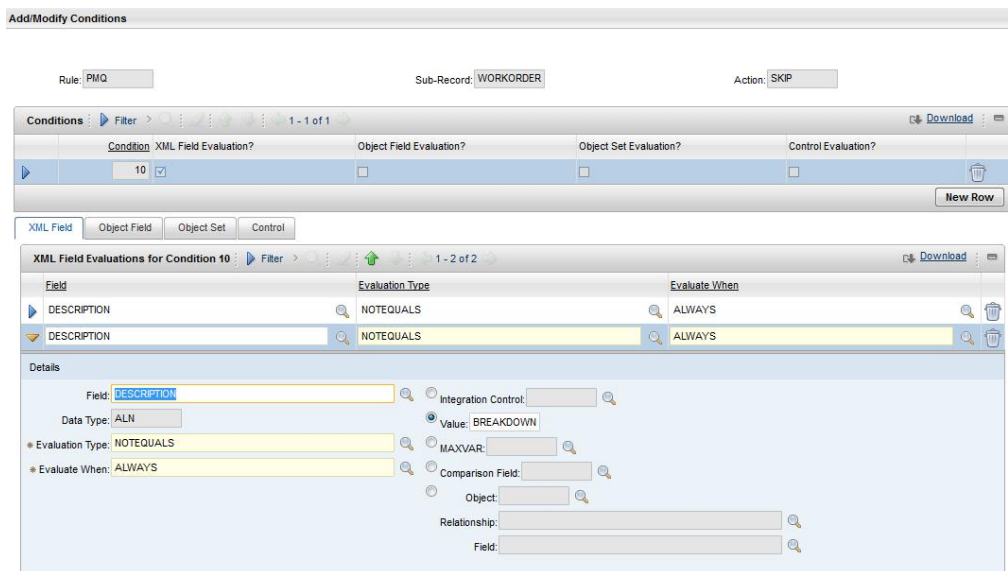


Figure 10. Adding a condition to skip the BREAKDOWN work order

3. Configure the external system to associate the publish channel and endpoint to the external system for the XML export of work orders.
 - a. Select **GoTo > Integration > External Systems**.

- b. Filter on SPEXTSYSTEM.
 - c. Select **Publish channels filter**.
 - d. Enter the following information:
 - In the **Publish Channel Name** field, type SPWO
 - In the **EndPoint** field, type MXXMLFILE
 - Enable the MXWOInterface for the external system SPEXTSYSTEM by selecting the **Enabled** check box.
 - Activate the external system (SPEXTSYSTEM) by selecting by selecting the **Enabled** check box.
- See the following figure.

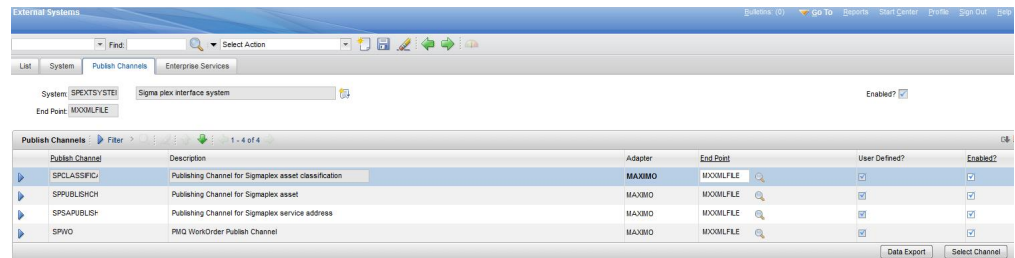


Figure 11. Enabling the external system SPEXTSYSTEM

Configuring Maximo to update recommendations in a work order

You can configure Maximo so that Maintenance work orders are updated in PMQ with PMQ recommendations.

The work order status is changed to CHANGED and Memo is updated to Refer LONGDESCRIPTION for PMQ recommendation. PMQ recommendation will get updated in the LONGDESCRIPTION field of PMQ.

The Maximo configuration described in this section creates the custom status CHANGED. The custom status CHANGED can be used to filter out all the work orders which were updated by PMQ with the recommendations.

Procedure

1. In Maximo, select **GoTo > System Configuration > Platform Configuration > Domains**.
2. Find the SYNONYM domain WOSTATUS to which you want to add a synonym value.

See the following figure.

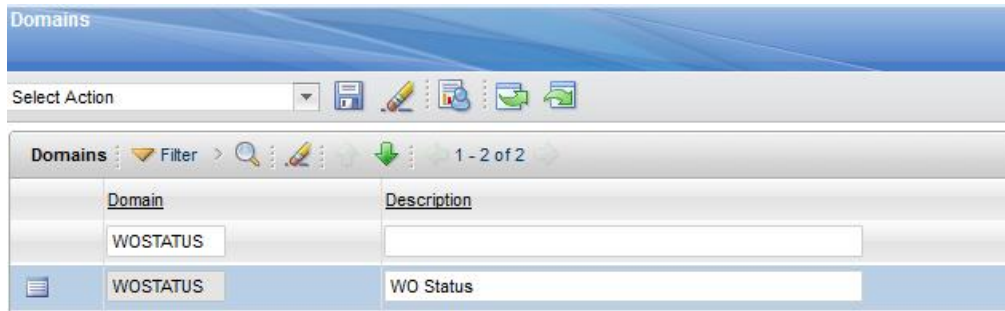


Figure 12. Finding the SYNONYM domain WOSTATUS

3. Click the **Edit details** icon.
4. Select **New Row** and specify the following values:
 - In the **Internal Value** field, specify WAPPR.
 - In the **Value** field, specify Change.
 - In the **Description** field, specify Recommendation Updated.
 See the following figure.

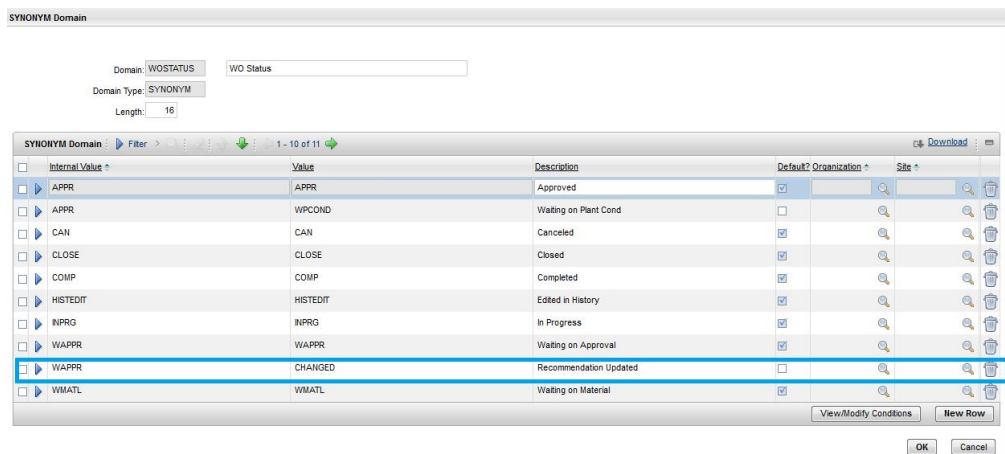


Figure 13. Specifying the values of the new row

Viewing work orders updated with PMQ recommendations

You can view work orders that have been updated with recommendations from IBM Predictive Maintenance and Quality.

Procedure

1. Select **Goto > Work Orders > Work Order Tracking**.
2. Select **Filter** and, in the **STATUS** field, specify **CHANGED**.
3. Open the work order and select the **Long description** button in the **Work Order** row.

See the following figure.

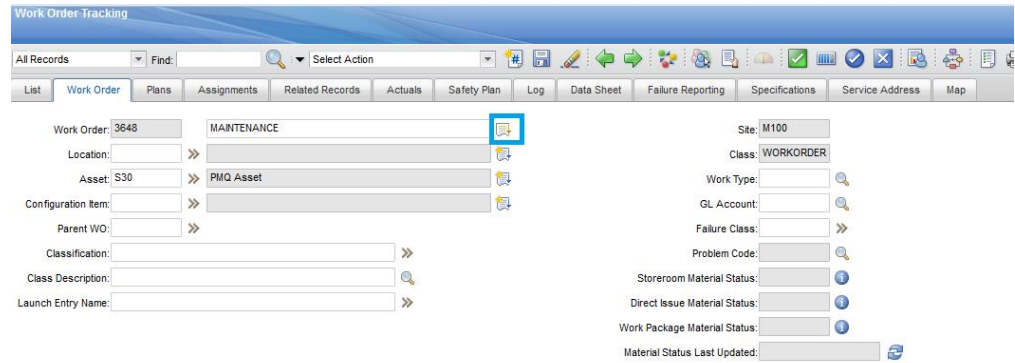


Figure 14. Opening the Long Description window

The PMQ recommendation appears, as shown in the following figure.

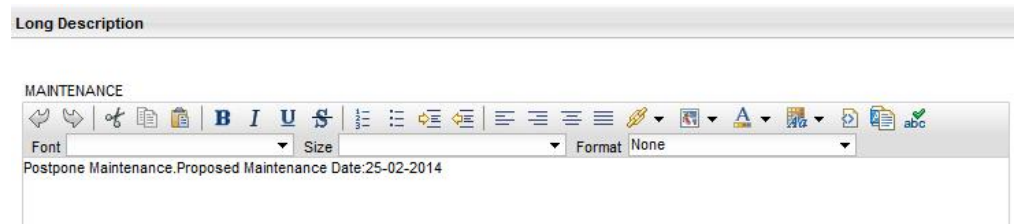


Figure 15. Viewing the PMQ recommendation

Creating a workorder in Maximo

You can create a MAINTENANCE workorder or a BREAKDOWN workorder in Maximo.

Procedure

1. Select **Goto > WorkOrders > Work Order Tracking > New Work Order**.
2. Specify the following values:
 - In the **Description** field, specify either BREAKDOWN or MAINTENANCE.
 - In the **Site** field, specify the Model No of the resource.
 - In the **Asset** field, specify the Serial No of the resource.
 - In the **Service Address** field, specify the location.
3. If you are creating a MAINTENANCE workorder, specify the following values:
 - In the **Scheduled Start** field, specify the scheduled maintenance start timestamp.
 - In the **Actual Start** field, specify the actual maintenance start timestamp, if applicable.
4. If you are creating a BREAKDOWN workorder, specify the following values:
 - In the **Reported Date** field, specify the Breakdown timestamp.

Results

For an example of a BREAKDOWN workorder, see the following figure.

Figure 16. Creating a BREAKDOWN work order

Mapping work orders for maintenance

You can map IBM Predictive Maintenance and Quality (PMQ) events to work orders for maintenance.

There are two types of work orders you can use for maintenance:

- Maintenance work orders
- Breakdown work orders

Mapping PMQ events to a Maintenance work order

Two PMQ events are generated from a maintenance work order: one event for scheduled maintenance (SM) and one event for actual maintenance (AM).

The event mapping is shown in the following table

Table 10. Mapping PMQ events to a Maintenance work order

PMQ event	Work Order	Remarks
incoming_event_cd	WONUM	
event_type_cd		Hardcoded to "MAINTENANCE"
source_system_cd		Hardcoded to "MAXIMO"
process_cd		
production_batch_cd		
location_cd	WOSERVICEADDRESS. SADDRESSCODE	
event_start_time	Scheduled Start	Timestamp field
event_end_time		
event_planned_end_time		
tenant_cd		Hardcoded to "PMQ"
operator_cd		
Model	SITEID	

Table 10. Mapping PMQ events to a Maintenance work order (continued)

PMQ event	Work Order	Remarks
serial_no	ASSETNUM	
measurement_type_cd		Hardcoded to "SM" for scheduled maintenance event and "AM" for actual maintenance
observation_timestamp	Scheduled Start for scheduled maintenance Actual Start for actual maintenance	Timestamp field
value_type_cd		Hardcoded to "ACTUAL"
observation_text	DESCRIPTION_ LONGDESCRIPTION	
Measurement		
material_cd		
multirow_no		Hardcoded to 1

Mapping PMQ events to a Breakdown work order

The event mapping is shown in the following table

Table 11. Mapping PMQ events to a Breakdown work order

PMQ event	Work Order	Remarks
incoming_event_cd	WONUM	
event_type_cd		Hardcoded to "MAINTENANCE"
source_system_cd		Hardcoded to "MAXIMO"
process_cd		
production_batch_cd		
location_cd	WOSERVICEADDRESS. SADDRESSCODE	
event_start_time	Reported date	Timestamp field
event_end_time		
event_planned_end_time		
tenant_cd		Hardcoded to "PMQ"
operator_cd		
Model	SITEID	
serial_no	ASSETNUM	
measurement_type_cd		Hardcoded to "BREAKDOWN"
observation_timestamp	Reported date	Timestamp field
value_type_cd		Hardcoded to "ACTUAL"

Table 11. Mapping PMQ events to a Breakdown work order (continued)

PMQ event	Work Order	Remarks
observation_text	DESCRIPTION_ LONGDESCRIPTION	
measurement		
material_cd		
multirow_no		Hardcoded to 1

Migrating historical work orders from Maximo to PMQ

You can migrate historical work orders from Maximo to PMQ using the following process:

1. Perform a manual export of the work orders in Maximo.
2. In PMQ, import the work orders on the ESB node.
3. Work orders with a description of MAINTENANCE or BREAKDOWN are mapped with PMQ events and loaded into PMQ DataStore through a file processing flow.

Note: Loading historical work orders is a onetime activity.

Migrating real-time work orders from Maximo to PMQ

You can migrate real-time work orders from Maximo to PMQ using the following process:

1. In Maximo, a new work order is created with the description MAINTENANCE or BREAKDOWN.
2. A web service is invoked from Maximo to IBM Integration Bus (IIB).
3. When the work order is updated with the maintenance date, the web service sends the work order details to PMQ in the form of a SOAP XML message.
4. The SOAP message is mapped to PMQ events and loaded into PMQ DataStore.

Chapter 5. Event data

Event data is any data that you want to measure about an event. Data comes from many sources, and it must be transformed into a format that can be used by IBM Predictive Maintenance and Quality.

For example, if the event is recording an inspection result, you might want to record: who was the inspector, when did the event take place, which product lot was it based on, and what was the result of the inspection?

IBM Integration Bus transforms data into a format that can be used in IBM Predictive Maintenance and Quality.

IBM Integration Bus has a visual interface that you use to map the data structure of the source data into the expected format.

The loading of event data involves the following steps:

1. In IBM Integration Bus, define the content and format of the event information that is coming in.
2. Map the data into the format that is expected by IBM Predictive Maintenance and Quality. You can use the graphical mapper, or for more complicated mappings, you can use a programming language such as Java™.
3. A message flow is provided to load data from a file. To use this flow, specify the file and location, and set a predefined time interval for checking the location. The file can be in a comma separated value format, for more information, see “File format and location” on page 24. However, by modifying a message flow, other formats such as XML are supported.

The data is processed:

- The data structure is brought into the correct format, then ported into event tables in the data store.
- The KPI and profile tables are calculated. KPIs are used in predictive models or in reports.
- This information is used to call a scoring service to receive a recommendation that is based on the current state of the event.
- The predictive model to be used is defined.

For information about the file locations and names and file format, see “File format and location” on page 24.

How events are processed

You must connect event sources to IBM Predictive Maintenance and Quality to enable events to be processed.

Events are processed in IBM Integration Bus and stored in the database. The database has an event store to record events, tables for key performance indicators (KPIs), and profiles that are related to the event source. KPIs provide a history of performance over time. Profiles show the current state of the event, and also include recommended actions from predictive models. Profiles help to speed up scoring.

The following steps occur:

1. IBM Integration Bus receives the events and maps them into the format that is required by IBM Predictive Maintenance and Quality with a custom flow, if required.
2. Events go into a queue (PMQ.EVENT.IN) for further processing, either as single events or multiple events that are processed together for efficiency.
3. Processed events are inserted into the event store. The information in the events immediately updates KPIs for the current KPI period. A historical record of the KPI values for each period is maintained (a period is typically one day). The event data is also used to immediately update profiles, which contain information about the current state of the event source.

This diagram shows the flow of events into IBM Integration Bus and into the database.

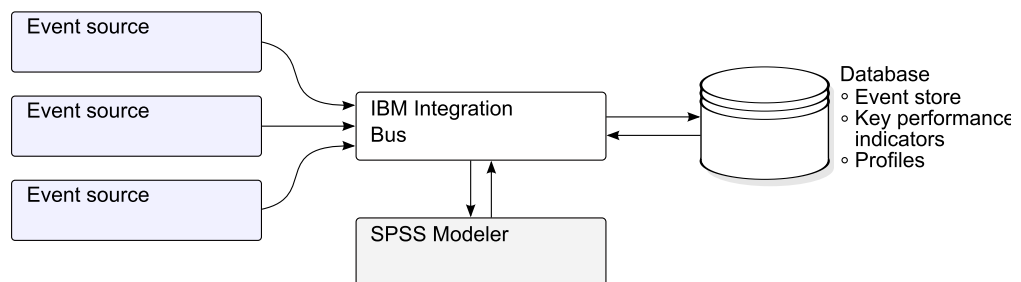


Figure 17. Flow of events into Integration Bus and into the database

The values in the event, KPI, and profile tables can be used as input to a predictive statistical model to generate recommended actions.

Processing events as they arrive, and immediately updating the aggregated values in the KPI and profile tables means that dashboards and reports are quickly updated with aggregated data.

Events must be loaded in chronological order. If events are not loaded in order then resulting KPIs and profiles may not be correct.

Event definition

Events are stored in the `event` and `event_observation` tables. An event can contain one or more event observations. Resource information is stored in the `event` table using `Resource_cd1` and `Resource_cd2`.

The calculated key performance indicators (KPIs) are stored in the `process_kpi` and `resource_kpi` tables. Event observations update the values in the `process_kpi` and `resource_kpi` tables.

The calculated profile values are stored in the `process_profile`, `resource_profile`, and `material_profile` tables. The values in the row are updated as the events arrive. The tables contain values for the current period (day), prior period (prior day), and lifetime to date.

KPIs are calculated at the day level.

Flat file event input

Events can be in a flat file format (.csv) or .xml format that must conform to the format required by IBM Predictive Maintenance and Quality. Events can be in other forms, such as web services; however IBM Integration Bus flows must be modified and extended.

Each event contains information recorded by one or more measurements or observations. An event can be associated with one or more materials. Each event can also have an associated operator and or device.

However, each row of the input file can only define a single event, a single material, a single operator, and a single device. Therefore an event containing more than one of these must have more than one row.

The values supplied for `material_cd` associate these materials with the event.

An event requiring more than one observation row must set the optional `multi_row_no` to 1 in the first row of the event. Additional rows must be directly beneath this row and increase the value set in `multi_row_no` by 1 for each additional row.

If `Resource_cd1` has a value and `Resource_cd2` is blank or null, then this event must be associated with an Agent or Operator. If `Resource_cd1` and `Resource_cd2` both have non-blank values and have rows in the `Master_Resource` table with `Resource_type` being ASSET, then they are termed as events from a device or a resource.

Each row of a multi-row event usually has a different observation. The columns marked as observation in the following table have different values in each row of a multi-row event.

Ensure that events are pre-mapped into this format to enable them to be loaded via the application programming interface (API).

In the following table, the first ten fields, `incoming_event_cd` to `tenant_cd` are common to all of the rows of a multi-row event. Only the values in the first row are used. Many of these field are codes which reference values in the master data tables. See Appendix B, "The flat file API," on page 175.

Table 12. Fields in the Events table

Field	Type	Optional or required	Event or observation	Description
<code>incoming_event_cd</code>	string(50)	optional	event	A unique code that identifies the event.
<code>event_type_cd</code>	string(50)	required	event	The event type, such as measurement, alarm, inspection.
<code>source_system_cd</code>	string(50)	optional	event	The system generating the event.
<code>process_cd</code>	string(50)	optional	event	The production process related to the event.
<code>production_batch_cd</code>	string(50)	optional	event	The production batch related to the event.

Table 12. Fields in the Events table (continued)

Field	Type	Optional or required	Event or observation	Description
location_cd	string(50)	optional	event	The location of the event.
event_start_time	datetime	required	event	Time the event started in Coordinated Universal Time (UTC) format, for example 2002-05-30T09:30:10-06:00.
event_end_time	datetime	optional	event	Time the event ended in UTC format.
event_planned_end_time	datetime	optional	event	Time the event was planned to end in UTC format.
tenant_cd	string(50)	optional	event	The organization associated with the event.
Resource_cd1	string(50)	optional	event	The operator associated with the event.
Resource_cd2	string(50)	optional	event	The model number of the device associated with the event.
Resource_cd1	string(50)	optional	event	The serial number of the device associated with the event.
measurement_type_cd	string(50)	required	observation	The measurement type determines how the event observation will be processed.
observation_timestamp	datetime	required	observation	The time associated with the observation in UTC format.
value_type_cd	string(50)	optional	observation	The type of observation (actual, planned, or forecast).
observation_text	string(400)	optional (see note)	observation	A description associated with the event.
measurement	float	optional (see note)	observation	A measurement associated with the event.
material_cd	string(50)	optional	observation	The material used for an event.
multirow_no	integer	optional		For multiple row events (more than one observation) use 1 to n for each row of the event.

Note: Either measurement or observation_text is required.

Schema definition of the event format

Events are processed in the event format that is shown in the following diagram. If you are extending IBM Predictive Maintenance and Quality to process external events from other sources, you must map those events to this internal event format.

The event schema is stored in the project `PMQEventDataLibrary`.

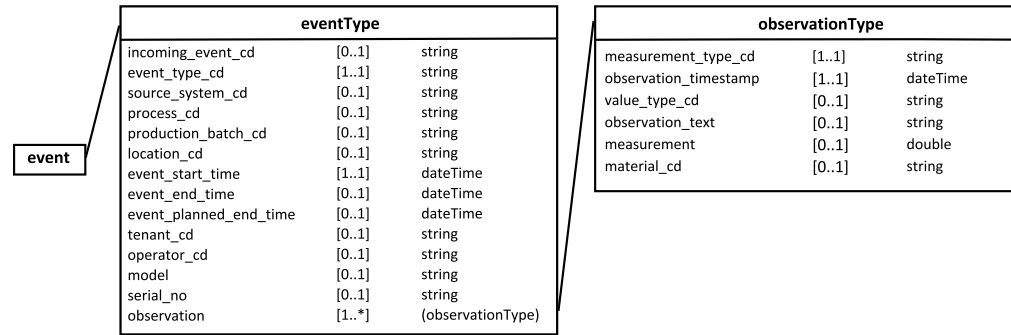


Figure 18. The event format used by IBM Predictive Maintenance and Quality

Error reporting

Errors can occur while processing events; during mapping to the required format or during the updating of the event, KPI, and profile tables.

You can add extra properties to the message to provide event source information for reporting while mapping to the IBM Predictive Maintenance and Quality format.

Profile and KPI tables

In addition to the event store and the master data, the IBM Predictive Maintenance and Quality database includes profile and KPI tables. The content of these tables is determined by a metadata driven aggregation mechanism that determines what calculations are performed when an event is processed.

The `measurement_type` and the `resource_type` or `material_type` values associated with the event and a particular `event_observation`, form the key that is used to look up the metadata.

Profile variables

The `profile_variable` table drives event processing in IBM Predictive Maintenance and Quality.

When an `event_observation` value arrives, its associated `measurement_type` value, and its associated `resource_type` value are used to find all of the `profile_variable` rows that are relevant for that observation, according to the orchestration defined for the event. Each of these rows indicates a calculation, which must be performed for the event. The calculation updates rows in the `kpi` and `profile` tables as indicated by the `profile_variable`. IBM Predictive Maintenance and Quality implements a standard set of calculations, but you can add a custom calculation and refer to it in a `profile_variable` row. The standard set of calculations include the following calculations:

- Measurement of type count

- Measurement text contains count
- Interval calculation
- Measurement above limit
- Measurement below limit
- Measurement delta

These calculations are described in “Profile calculations” on page 59.

To be able to process some events, you must load mandatory profile variables and measurement types. For more information, see “Mandatory profile variables and measurement types” on page 192.

For example, a temperature event with the measurement_type value “Ambient temperature” from a device can be aggregated by defining a profile_variable for the measurement_type “Ambient temperature” with the profile_calculation “Measurement of type” and adding a profile update for the measurement_type to the orchestration. This causes a row to be added to the resource_kpi table at each period for this device and profile_variable. This row aggregates the temperature values across each period (day). In addition, the defined profile_variable causes a row to be added to the resource_profile table for this device that is updated as each temperature event is processed.

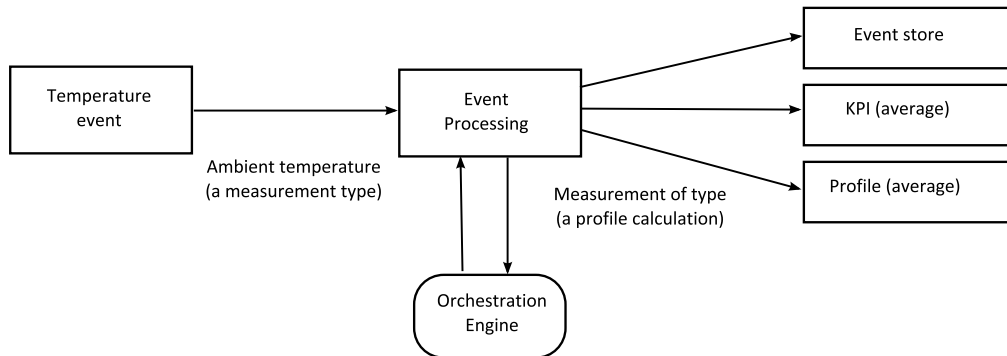


Figure 19. Temperature event workflow

Making a profile variable inactive

To make a profile variable inactive, for example, if you want to prevent a calculation from being performed, remove the profile update from the orchestration.

KPI tables

The IBM Predictive Maintenance and Quality key performance indicator (KPI) tables: resource_kpi and process_kpi hold aggregated values for each day.

In the resource_kpi table, the key for each row is determined by

- The profile_variable that triggered the calculation of the KPI
- The date
- The resource that is associated with the event
- The event code that is associated with the event observation
- The location that is associated with the event
- The process that is associated with the event

- The production batch that is associated with the event
- The tenant_id.

The fields in resource_kpi are described in the following table.

Table 13. Fields in the resource_kpi table

Field	Type	Description
kpi_date	date	The date for which the KPI is calculated. The time grain for KPI calculation is a single day.
profile_variable_id	integer	The profile variable that is the source of this KPI.
resource_id	integer	The resource that is associated with the event.
event_code_id	integer	The event code that is associated with the event observation. Event codes are codes for alarms, failures, issues, and so on. When an event arrives with a measurement_type value that has an event_code_indicator value of 1, the text from the event_observation_text field is assumed to contain an event_code value.
location_id	integer	The location that is associated with the event.
process_id	integer	The process that is associated with the event.
production_batch_id	integer	The production batch that is associated with the event.
actual_value	float	The actual value for this KPI. It is important to understand that for Business Intelligence reporting purposes, this value is typically divided by the measure count. Even if the value is meant to be an average, this value must be a sum of the values from the event and the measure_count must be the number of events. actual_value field supports the average calculation for dimensional reporting.
plan_value	float	The planned value for the KPI for this date.
forecast_value	float	The forecast value for the KPI for this date
measure_count	integer	The measure count for this date. Typically this value is used to divide the actual_value for reporting.
current_indicator	integer	Indicates that this row is the current row for a KPI. Typically the date of the current row is the current day.
tenant_id	integer	The tenant_id of the profile_variable that is the source of this KPI.

The fields in the process_kpi table are described in the following table.

Table 14. Fields in the process_kpi table

Field	Type	Description
process_id	integer	The process that is associated with the resource.
kpi_date	date	The date for which the KPI is calculated. The time grain for KPI calculation is a single day.
profile_variable_id	integer	The profile variable that is the source of this KPI.
material_id	integer	The material that is associated with the resource.

Table 14. Fields in the process_kpi table (continued)

Field	Type	Description
event_code_id	integer	The event code that is associated with the event observation. Event codes are codes for alarms, failures, issues, and so on. When an event arrives with a measurement_type value that has an event_code_indicator value of 1, the text from the event_observation_text field is assumed to contain an event_code value.
location_id	integer	The location that is associated with the resource.
production_batch_id	integer	The production batch that is associated with the event.
actual_value	float	The actual value for this KPI. It is important to understand that for Business Intelligence reporting purposes, this value is typically divided by the measure count. Even if the value is meant to be an average, this value must be a sum of the values from the resource and the measure_count must be the number of resources. actual_value field supports the average calculation for dimensional reporting.
plan_value	float	The planned value for the KPI for this date.
forecast_value	float	The forecast value for the KPI for this date.
measure_count	integer	The measure count for this date. Typically this value is used to divide the actual_value for reporting.
current_indicator	integer	Indicates that this row is the current row for a KPI. Typically the date of the current row is the current day.
tenant_id	integer	The tenant_id of the profile_variable that is the source of this KPI.

Profiles

Profiles provide pre-aggregated values to enable near real-time display in reports and dashboards.

The fields in the resource_profile table are described in the following table.

Table 15. Fields in the resource_profiles table

Field	Type	Description
resource_id	integer	The resource associated with this profile.
profile_variable_id	integer	The profile_variable that is the source of this profile.
value_type_id	integer	Value type of this profile. One of actual, plan, and forecast.
event_code_id	integer	The event code associated with the event observation. These are codes for alarms, failures, issues, and so on. When an event arrives with a measurement_type having an event_code_indicator of 1, the text from event_observation_text is assumed to contain an event_code.
location_id	integer	The location associated with the event.

Table 15. Fields in the `resource_profiles` table (continued)

Field	Type	Description
<code>profile_date</code>	datetime	This date is based on the timestamp of the most recent event used to update the profile.
<code>last_profile_date</code>	datetime	
<code>period_average</code>	float	The average value for the period.
<code>period_min</code>	float	The minimum value for the period.
<code>period_max</code>	float	The maximum value for the period.
<code>period_total</code>	float	The total value for the period.
<code>period_std_dev</code>	float	The standard deviation for the period.
<code>period_msr_count</code>	integer	The number of events contributing to this profile for the current period.
<code>prior_average</code>	float	The average value for the prior period.
<code>prior_min</code>	float	The minimum value for the prior period.
<code>prior_max</code>	float	The maximum value for the prior period.
<code>prior_total</code>	float	The total value for the prior period.
<code>prior_std_dev</code>	float	The standard deviation for the prior period.
<code>prior_msr_count</code>	integer	The number of events contributing to this profile for the prior period.
<code>ltd_average</code>	float	The average value lifetime to date.
<code>ltd_min</code>	float	The minimum value lifetime to date.
<code>ltd_max</code>	float	The maximum value lifetime to date.
<code>ltd_total</code>	float	The total value lifetime to date.
<code>ltd_std_dev</code>	float	The standard deviation lifetime to date.
<code>ltd_msr_count</code>	integer	The number of events contributing to this profile for the lifetime to date.
<code>last_value</code>	float	The most recent value in <code>event_observation.measurement</code> that updated this profile.
<code>tenant_id</code>	integer	The <code>tenant_id</code> of the <code>profile_variable</code> that is the source of this KPI.

Profile calculations

Profile calculations update the key performance indicator (KPI) and profile table (the `kpi_indicator` and `profile_indicator` values are updated). A profile variable specifies the profile calculations to perform for an observation with a given measurement type.

A profile variable maps a measurement type to a profile calculation. There may be zero or more profile variables for a given measurement type.

The following section describes the default profile calculations.

Note: Not all of the profile calculations are covered. Only the profile calculations used by BI and Analytics are covered as a part of Foundation porting.

Measurement of type

This calculation is based on the value of a particular `measurement_type`.

- KPI: the `actual_value` column contains the sum of all the `event_observation.measurement` values. The `measure_count` column contains a count of all the `event_observation` events.
- Profile: the average, minimum, maximum, total, and standard deviations are calculated for the present, prior (previous day), and lifetime to date periods. The average value in the profile is the true average and, unlike KPI, is not divided by the corresponding `msr_count` value. These values can be calculated on a running basis for efficiency. The `msr_count` values record the count of all the `event_observation` events in the period. The `last_value` column contains the most recent `event_observation.measurement` values.

Measurement of type count

A count of the number of times an event with a particular `measurement_type` occurs.

- KPI: the `actual_value` and `measure_count` columns contain a count of the occurrences of the specified `event_observation`.
- Profile: the `msr_count` values record the count of the `event_observation` events in the period.

Measurement text contains count

A count of the number of times an event observation text contains a string. The string is the value of the `profile_variable.comparison_string`.

- KPI: the `actual_value` and `measure_count` columns contain a count of the occurrences of the specified `event_observation` events.
- Profile: the `msr_count` values record the count of the `event_observation` events in the period.

Measurement above limit

This is a count of the number of times the `event_observation.measurement` value falls above the value of the profile variable (`high_value_number`).

- KPI: the `actual_value` and `measure_count` columns contain a count of the occurrences of the specified `event_observation`.
- Profile: the `msr_count` values record the count of the `event_observation` events in the period.

Measurement below limit

This is a count of the number of times the `event_observation.measurement` value falls below the value of the profile variable (`low_value_number`).

- KPI: the `actual_value` and `measure_count` columns contain a count of the occurrences of the specified `event_observation` events.
- Profile: the `msr_count` values record the count of the `event_observation` events in the period.

Measurement delta

This is the change from one measurement value to the next.

- KPI: the `actual_value` column contains a sum of all the changes in measurement values. The `measure_count` column contains a count of all the `event_observation` events.
- Profile: The `msr_count` value should be 1 if the `event_observation` event occurs in the period. The `profile_date` value has the timestamp of the most recent `event_observation` event.

Custom calculations

You can modify the event processing flow to support extra calculations.

The custom calculation must be defined in the solution definition file. The custom calculation must be implemented as a Java Class implementing the `com.ibm.analytics.foundation.calculation.api.Calculation`.

Predictive scoring

To provide a health score for predictive models, code is required in the event processing flow.

A scoring service requires a defined set of inputs and returns a result. The score returns either a numerical value, a recommendation, or both. The sources of data for the input to the scoring service are the Event, KPI (key performance indicator), and Profile tables. Code transforms the data that is needed to supply the exact set of input parameters that are required by the scoring service. The scoring service is called by a web service call from IBM Integration Bus.

When the results are returned from the scoring service, they are written back as new events. Measurement types and profile variables can be defined for these events.

For example, a health score and recommendation can be recorded as an `event_observation.measurement` and `event_observation.observation_text`. In addition to being stored in the event tables, this score and recommendation can be aggregated for IBM Cognos Business Intelligence Reporting by defining two `profile_variables` and the corresponding profile updates in the profile adapter configuration of an orchestration.

To aggregate the health score, define a `profile_variable` and `profile_adapter` configuration for the Measurement of type calculation.

To aggregate the occurrences of a specific recommendation, one needs to define a `profile_variable` and `Profile_adapter` configuration for a Text contain calculation and set the `comparison_string` attribute of `profile_variable` and `profile_adapter` to the name of the recommendation.

The processing of an event that contains the result of a predictive scoring service can invoke a second scoring service.

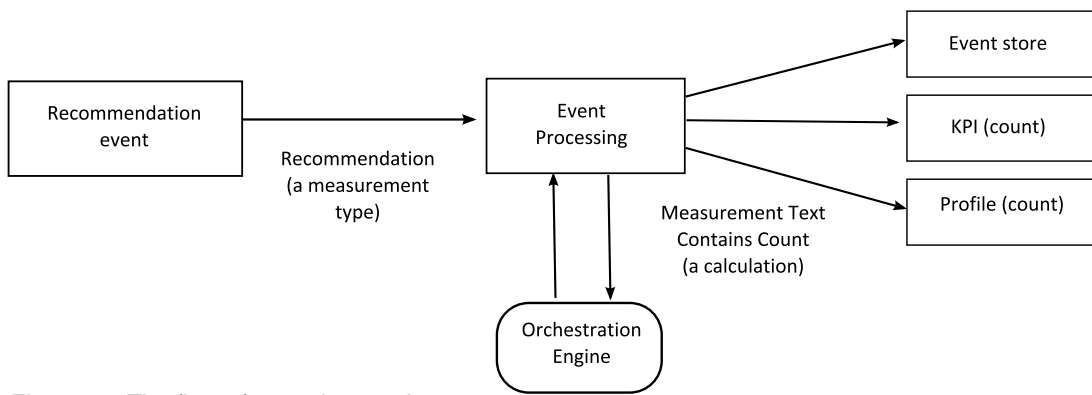


Figure 20. The flow of a scoring service

How scoring is triggered

Scoring for predictive models is triggered based on the service adapter configuration defined in the orchestration xml file. To construct any custom scoring, the orchestration xml file must be defined accordingly.

Events and actual, planned, and forecast values

Typically, events contain actual values. Special events may contain planned values, and forecast values.

At least one event containing planned or forecast values, should be supplied for each KPI reporting period (day). This allows planned and forecast values to appear on IBM Cognos Business Intelligence reports along with actual values.

Event processing queue

Two queues are used to gather events for processing. One queue is for events read from .csv files, or transformation flows that you have developed. The other queue is for events that are generated from scoring results. You can use additional queues for processing but only one queue can contain events that update the same Key Performance Indicators (KPI) or profile rows. Typically, a queue supports events from an exclusive set of resources or processes.

A queue is used to hold events for processing in a single thread. The queue only contains events already mapped to the IBM Predictive Maintenance and Quality format.

Event processing

Event processing consists of the following steps.

1. Lookup of primary keys for supplied business keys.
2. Insertion of events.
3. Updating and inserting the KPI and Profile rows.
4. Scoring using an IBM SPSS predictive model.
5. Making a recommendation using IBM Analytical Decision Management.
6. Work order creation.

Recording and acting on predictive scores and recommendations

Profile variables are used to determine what key performance indicators (KPI) and profile calculations must be performed for an event. However, profile variables do not determine whether scoring or decision management is performed for an event. Scoring or decision management is determined by the service adapter definition in the Orchestration XML. This Orchestration XML must be modified to provide customization in scoring and decision making.

Scores that are returned by a predictive model and recommendations that are returned by decision management are processed and recorded in the same way as events that are received from a device. This means that scores and recommendation results are written to a content store, KPIs and profiles are calculated for these values, and the values are displayed in reports.

This reuse of the event processing mechanism is implemented by creating an event that uses the standard event format. An appropriate event type and measurement type are used for the event. The event is further processed, based on the service adapter definition defined in the orchestration xml file. Events on this internal event processing queue are processed by the same flow as external events. Profile variables and profile updates in the profile adapter configuration are defined to control the processing of these internal events to calculate KPI and Profile values.

If IBM Predictive Quality and Maintenance is configured to work with IBM Maximo Asset Management, a recommendation can result in the creation of a work order in IBM Maximo. Customizing this behavior also requires modifying ESQL code.

For more information, see Chapter 9, “Recommendations,” on page 145

Threads

Events are processed by only one flow that runs in a single thread. If more than one flow is implemented to process events, these different flows must not update the same KPI or Profile rows. A single thread is required to ensure that only a single thread is calculating and updating a row in the KPI and Profile tables.

Batch processing

Event processing can occur faster by processing more than one event at the same time through batch processing. For example, if you want to process and load event data for one year, then you can do so by processing the events through multiple .csv files.

Use this approach only if the separate files contain events from separate devices.

- Create copies of `MultiRowEventLoad` flow and deploy on the broker. Each copy of message flow processes one .csv file at a time.
- Ensure that you do not set the `AdditionalInstances` property of the `MultiRowEventLoad` flow to greater than 0 for processing the batch concurrently.
- Ensure that the events from the same resource are combined into a single file in chronological order.

Parallel processing

Event processing can also occur faster by processing more than one event at the same time. However, it is important that only one thread at a time updates a row in the KPI or profile tables. Since the rows in these tables are related to resources and measurement types, achieve thread isolation by ensuring that events from an individual resource or of a particular measurement type are processed by a single thread. You can implement parallel processing by using multiple queues to manage the separation of events.

Event processing assumes that only one thread updates an individual row in the `resource_kpi`, `resource_profile`, `process_kpi`, `process_profile`, and `material_profile` tables. This is true for events from external devices and internal events that record recommendations. This means that parallelism can be achieved only by segmenting the events into groups that do not share resources, processes, or materials. To achieve parallelism, you must deploy multiple copies of event and integration flows, and ensure that each copy of the message flow uses a unique set of queues.

Remove events

Normally events are not deleted from the analytic database. During testing and development, events can be removed by deleting the appropriate rows from the `event`, `event_observation`, and `event_resource` tables.

As events are processed, extra internal events are added when predictive scoring and decision management are performed. You can also remove these events.

Sample remove event code

The following SQL code is an example and must be modified.

```
DELETE FROM SYSREC.EVENT_RESOURCE ER WHERE...
DELETE FROM SYSREC.EVENT_OBSERVATION EO WHERE...
DELETE FROM SYSREC.EVENT E WHERE...
```

Event processing also adds row to the KPI and profile tables, and you can remove those rows by modifying the following SQL.

```
DELETE FROM SYSREC.RESOURCE_KPI RK WHERE...
DELETE FROM SYSREC.RESOURCE_PROFILE RP WHERE...
DELETE FROM SYSREC.PROCESS_KPI PK WHERE...
DELETE FROM SYSREC.PROCESS_PROFILE PP WHERE...
DELETE FROM SYSREC.MATERIAL_PROFILE MP WHERE...
```

Configuring solution.xml for the event flow

The event definition, like the master data definition, is part of the solution XML file.

Within the `solution.xml` for event handling, there is one xml structure for a table where both `event` and `event_observation` is covered. The `event_resource` used in PMQ 1.0 is removed by having the `resource_information` defined within the event xml. Within the event definition, there is a separate tag called `observation` with the `table_cd` element.

```
<event_definition>
  <table table_cd="EVENT">
    <column column_cd="EVENT_START_TIME" type="timestamp" />
    <column column_cd="EVENT_END_TIME" type="timestamp" is_nullable="true" />
    <column column_cd="EVENT_PLANNED_END_TIME" type="timestamp" is_nullable="true" />
  </table>
</event_definition>
```

```

<column column_cd="INCOMING_EVENT_CD" type="string" size="200" is_nullable="true" />
<reference reference_cd="ASSET_ID" table_reference="MASTER_RESOURCE">
  <column_mapping reference_column_cd="SERIAL_NO" table_column_cd="RESOURCE_CD1"/>
  <column_mapping reference_column_cd="MODEL" table_column_cd="RESOURCE_CD2"/>
</reference>
<reference reference_cd="AGENT_ID" table_reference="MASTER_RESOURCE">
  <column_mapping reference_column_cd="OPERATOR_CD" table_column_cd="RESOURCE_CD1"/>
  <column_mapping reference_column_cd="OPERATOR_NA" table_column_cd="RESOURCE_CD2"/>
</reference>
<reference reference_cd="EVENT_TYPE_ID" table_reference="MASTER_EVENT_TYPE" />
<reference reference_cd="SOURCE_SYSTEM_ID" table_reference="MASTER_SOURCE_SYSTEM" />
<reference reference_cd="PROCESS_ID" table_reference="MASTER_PROCESS" />
<reference reference_cd="PRODUCTION_BATCH_ID" table_reference="MASTER_PRODUCTION_BATCH" />
<reference reference_cd="LOCATION_ID" table_reference="MASTER_LOCATION"/>
<observation table_cd="EVENT_OBSERVATION">
  <column column_cd="OBSERVATION_TIMESTAMP" is_key="true" type="timestamp" />
  <column column_cd="OBSERVATION_TEXT" type="string" size="800" is_nullable="true" />
<column column_cd="MEASUREMENT" type="double" is_nullable="true"/>
  <reference reference_cd="MEASUREMENT_TYPE_ID" is_key="true" table_reference="MASTER_MEASUREMENT_TYPE" />
  <reference reference_cd="VALUE_TYPE_ID" is_key="true" table_reference="MASTER_VALUE_TYPE" />
  <reference reference_cd="EVENT_CODE_ID" is_key="true" table_reference="MASTER_EVENT_CODE"/>
  <reference reference_cd="MATERIAL_ID" table_reference="MASTER_MATERIAL"/>
  <event_interval_column column_cd="OBSERVATION_DATE" type="date" />
  <event_interval_column column_cd="OBSERVATION_TIME" type="time" />
</observation>
</table>
</event_definition>

```

For handling the resource related information there are two references defined in the event xml.

```

<reference reference_cd="ASSET_ID" table_reference="MASTER_RESOURCE">
  <column_mapping reference_column_cd="SERIAL_NO" table_column_cd="RESOURCE_CD1"/>
  <column_mapping reference_column_cd="MODEL" table_column_cd="RESOURCE_CD2"/>
</reference>
<reference reference_cd="AGENT_ID" table_reference="MASTER_RESOURCE">
  <column_mapping reference_column_cd="OPERATOR_CD" table_column_cd="RESOURCE_CD1"/>
  <column_mapping reference_column_cd="OPERATOR_NA" table_column_cd="RESOURCE_CD2"/>
</reference>

```

If the referenced resource is a ASSET or AGENT.

The xml structure with in the event for handling the observation part is defined by a separate xml element called observation.

```

<observation table_cd="EVENT_OBSERVATION">
  <column column_cd="OBSERVATION_TIMESTAMP" is_key="true" type="timestamp" />
  <column column_cd="OBSERVATION_TEXT" type="string" size="800" is_nullable="true" />
<column column_cd="MEASUREMENT" type="double" is_nullable="true"/>
  <reference reference_cd="MEASUREMENT_TYPE_ID" is_key="true" table_reference="MASTER_MEASUREMENT_TYPE" />
  <reference reference_cd="VALUE_TYPE_ID" is_key="true" table_reference="MASTER_VALUE_TYPE" />
  <reference reference_cd="EVENT_CODE_ID" is_key="true" table_reference="MASTER_EVENT_CODE"/>
  <reference reference_cd="MATERIAL_ID" table_reference="MASTER_MATERIAL"/>
  <event_interval_column column_cd="OBSERVATION_DATE" type="date" />
  <event_interval_column column_cd="OBSERVATION_TIME" type="time" />
</observation>

```

Chapter 6. Quality early warning system use cases

The quality early warning system (QEWS) in IBM Predictive Maintenance and Quality 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.
- Shipping a large quantity of defective products to distribution channels or customers causes high warranty expenses.
- 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.

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*.

QEWS offers three use cases. *Quality inspection* detects unfavorable changes in the quality of components. *Warranty* detects warranty issues sooner than they might otherwise be detected, which results in fewer warranty claims and lower costs. *Parametric* detects unfavorable changes in variable-type data, and provides information that facilitates diagnostics and alarm prioritization.

Quality inspection

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.

The quality early warning system (QEWS) in IBM Predictive Maintenance and Quality 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 QEWS quality inspection use case 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)

The quality inspection solution is not connected only to products. It is also connected to resource, process, material, and location entities. References to these entities are included in the PRODUCT_KPI and PRODUCT_PROFILE tables so that 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.

Business and technical challenges

You need the best techniques to examine quality data from tens of thousands of products and to provide proactive quality management.

You need to be able to detect process variability that is not visible through traditional means such as trend analysis. QEWS can evaluate trace data and predict with a tunable confidence whether the variability in the data is natural “noise” or a subtle indication of an impending problem. This ability is a significant improvement over traditional statistical process control.

Business challenges

Better analytical methods are available but are difficult to implement. This is due to complex computational challenges and constraints in software implementation.

Technical challenges

Manufacturing process variations can occur slowly. Gradual changes in product quality are not detected or they are detected too late, which leads to a large inventory of suspicious or defective lots. This results in increased inspection effort, more low quality products, and more waste.

Defining the quality inspection solution

To define the quality inspection solution, you must load master data, load event data, define messages flows, and define the output location of the inspection analysis.

Procedure

1. Load master data. For more information about loading master data, see Chapter 4, “Master data,” on page 23.

2. Load event data. You can load event data in batch mode or in real time. For more information about loading event data, see Chapter 5, “Event data,” on page 51.
3. Define message flows. For more information about message flows, see “Message flows” on page 15.

Results

IBM Cognos Business Intelligence uses data in the PRODUCT_KPI and PRODUCT_PROFILE tables to generate the inspection dashboards and reports.

Quality inspection solution details

You must consider some requirements when you are loading the master data and event data tables.

Master data tables are loaded by the master flows. The following tables are required to implement an inspection use case:

Master_Event_Type

You must define the following event types in the Master_Event_Type table:

PRODUCTION

Defines the products that are produced by the process.

INSPECTION

Defines the sample set of products that are being inspected.

The following text is an example of a CSV file that is used to load the Master_Event_type table:

```
event_type_cd,event_type_name,language_cd,tenant_cd
PRODUCTION,PRODUCTION,EN,PMQ
INSPECTION,INSPECTION,EN,PMQ
```

Master_Value_Type

There are three possible values for value_type_cd in the Master_Value_Type table: ACTUAL, PLAN, FORECAST. Usually, the data that is associated with PRODUCTION or INSPECTION events is ACTUAL.

The following text is an example of a CSV file that is used to load the Master_Value_Type table:

```
value_type_cd,value_type_name,language_cd,tenant_cd
ACTUAL,Actual,EN,PMQ
PLAN,Plan,EN,PMQ
FORECAST,Forecast,EN,PMQ
```

Master_Location

The Master_Location table contains information that is specific to the location where the event occurs, or the resource that produces the event.

The following text is an example of a CSV file that is used to load the Master_Location table:

```
location_cd,location_name,region_cd,region_name,country_cd,
country_name,state_province_cd,state_province_name,city_name,latitude,
longitude,language_cd,tenant_cd,is_active
Tokyo,Tokyo,AP,Asia Pacific,JP,Japan,TY,Tokyo,TokyoCity, 35.41,139.45,
EN,PMQ,1
```

Master_Measurement_Type

The Master_Measurement_Type table defines how the observation is read or used. For inspection, the measurement_type is INSPECT and FAIL. The INSPECT measurement defines how many units of product were inspected or tested for quality. The FAIL measurement defines whether the outcome of the inspection is successful or not, which is identified by a flag with the FAIL.

The following text is an example of a CSV file that is used to load the Master_Measurement_Type table:

```
measurement_type_cd,measurement_type_name,unit_of_measure,
carry_forward_indicator,aggregation_type,event_code_indicator,language_cd,
tenant_cd
INSPECT,INSPECTION,,0,AVERAGE,0,EN,PMQ
FAIL,FAIL_QTY_INDICATOR,,0,AVERAGE,0,EN,PMQ
INSP_LAM0,Inspection Acceptance Level,,0,SUM,0,EN,PMQ
INSP_LAM1,Inspection Unacceptance Level,,0,SUM,0,EN,PMQ
INSPECT_NO_DAYS,Inspect No of days,,0,SUM,0,EN,PMQ
INSP_PROB0,Inspection Confidence Probability,,0,SUM,0,EN,PMQ
```

Parameter names are loaded as measurement type. Parameter names such as **LAM0**, **LAM1**, and **PROB0** are all considered for measurement type, because their values are loaded by using event formats.

Master_Product

The Master_Product table contains the core data for the inspection use case. This table stores information that is related to a product and the product_type.

The following text is an example of a CSV file that is used to load the Master_Product table:

```
product_cd,product_name,product_type_cd,product_type_name,language_cd,tenant_cd,
is_active
WT2444,Wind Turbine,Type Turbine,Type Turbine,EN,PMQ,1
Prd_No_1,Product Name 1,Type1,Type1,EN,PMQ,1
Prd_No_2,Product Name 2,Type2,Type2,EN,PMQ,1
Prd_No_3,Product Name 3,Type3,Type3,EN,PMQ,1
Prd_No_4,Product Name 4,Type4,Type4,EN,PMQ,1
Prd_No_5,Product Name 5,Type5,Type5,EN,PMQ,1
Prd_No_6,Product Name 6,Type6,Type6,EN,PMQ,1
Prd_No_7,Product Name 7,Type7,Type7,EN,PMQ,1
Prd_No_8,Product Name 8,Type8,Type8,EN,PMQ,1
Prd_No_9,Product Name 9,Type9,Type9,EN,PMQ,1
Prd_No_10,Product Name 10,Type10,Type10,EN,PMQ,1
```

Master_Production_Batch

The Master_Production_Batch table contains information about each production batch that is used to produce a product. The information includes the product that is produced, the date it is produced, and the batch information.

The following text is an example of a CSV file that is used to load the Master_Product table:

```
production_batch_cd,
production_batch_cd,production_batch_name,product_cd,product_type_cd,
produced_date,language_cd,tenant_cd
T1,Turbine,WT2444,Type Turbine,2010-01-01,EN,PMQ
T2,Turbine,WT2444,Type Turbine,2011-01-01,EN,PMQ
PB 1,Production Batch 1,Prd_No_1,Type1,2011-12-08,EN,PMQ
PB 2,Production Batch 2,Prd_No_2,Type2,2011-03-18,EN,PMQ
PB 3,Production Batch 3,Prd_No_3,Type3,2012-01-04,EN,PMQ
PB 4,Production Batch 4,Prd_No_4,Type4,2012-06-06,EN,PMQ
PB 12,Production Batch 12,Prd_No_4,Type4,2012-06-06,EN,PMQ
```



```

PB 5,Production Batch 5,Prd_No_5,Type5,2012-10-26,EN,PMQ
PB 6,Production Batch 6,Prd_No_6,Type6,2013-07-07,EN,PMQ
PB 7,Production Batch 7,Prd_No_7,Type7,2011-11-28,EN,PMQ
PB 8,Production Batch 8,Prd_No_8,Type8,2011-12-19,EN,PMQ
PB 9,Production Batch 9,Prd_No_9,Type9,2012-08-17,EN,PMQ

```

Master_Profile_Variable

The Profile Parameters table uses parameters across the Product, Production Batch, Resource, Material, Process, and Location master entities. The Profile Parameter table entries are built by using the Profile Variable Code as one of the keys, along with related master keys (mainly Product, with either or all other master keys, such as Resource, Process, Material, and Location). To start loading the Profile Parameters table, the Master_Profile_Variable table must be prepared. The convention that is used for the Profile Variable Code is INSP_ combined with **Parameter Name**. For example, for the parameter name LAM0, the Profile Variable Code is INSP_LAM0.

The following text is an example of a CSV file that is used to load the Master_Profile_Variable table:

```

profile_variable_cd,profile_variable_name,profile_calculation_name,
measurement_type_cd,resource_type_cd,material_type_cd,profile_units,comparison_string,
low_value_date,high_value_date,low_value_number,high_value_number,kpi_indicator,
profile_indicator,data_type,aggregation_type,carry_forward_indicator,
process_indicator,variance_multiplier,language_cd,tenant_cd
INSP_LAM0,Inspection Acceptance Level,ASSIGN,INSP_LAM0,ASSET,-NA-
,,,,,1,1,INT,,0,0,1,EN,PMQ
INSP_LAM1,Inspection Unacceptable Level,ASSIGN,INSP_LAM1,ASSET,-
NA-,,,,,1,1,INT,,0,0,1,EN,PMQ
INSPECT_NO_DAYS,Inspection No of days,ASSIGN,INSPECT_NO_DAYS,ASSET,-
NA-,,,,,1,1,INT,,0,0,1,EN,PMQ
INSP_PROB0,Inspection Confidence Probability,ASSIGN,INSP_PROB0,ASSET,-
NA-,,,,,1,1,INT,,0,0,1,EN,PMQ

```

Event data loading

In addition to normal batch events, parameter values are loaded by using events. Parameters are loaded first, followed by normal inspection-related events. The following text is an example of the event format that is used to load parameter values for inspection, supported by the event type PARAMETERVI.

```

incoming_event_cd,event_type_cd,source_system_cd,process_cd,prod_batch_cd,
location_cd,event_start_time,event_end_time,event_planned_end_time,tenant_cd,operator_cd,
model,serialNo,measurement_type_cd,observation_timestamp,value_type_cd,observation_text,
measurement,material_code,multirow_no
1,PARAMETERVI,,-NA-,PP9-XX9-009,-NA-,2014-12-02 00:51:35,2014-12-02 00:51:35,2014-12-02
00:51:35,PMQ,,-NA-,-NA-,INSP_LAM0,2014-12-02 00:51:35,ACTUAL,INSP_LAM0,5,-NA-,1
2,PARAMETERVI,,-NA-,PP9-XX9-009,-NA-,2014-12-02 00:51:35,2014-12-02 00:51:35,2014-12-02
00:51:35,PMQ,,-NA-,-NA-,INSP_LAM1,2014-12-02 00:51:35,ACTUAL,INSP_LAM1,8.5,-NA-,1
3,PARAMETERVI,,-NA-,PP9-XX9-009,-NA-,2014-12-02 00:51:35,2014-12-02 00:51:35,2014-12-02
00:51:35,PMQ,,-NA-,-NA-,INSPECT_NO_DAYS,2014-12-02
00:51:35,ACTUAL,INSPECT_NO_DAYS,2000,-NA-,1
4,PARAMETERVI,,-NA-,PP9-XX9-009,-NA-,2014-12-02 00:51:35,2014-12-02 00:51:35,2014-12-02
00:51:35,PMQ,,-NA-,-NA-,INSP_PROB0,2014-12-02 00:51:35,ACTUAL,INSP_PROB0,0.99,-NA-,1
5,PARAMETERVI,,-NA-,PPB-XXY-003,-NA-,2014-12-02 00:51:35,2014-12-02 00:51:35,2014-12-02
00:51:35,PMQ,,-NA-,-NA-,INSP_LAM0,2014-12-02 00:51:35,ACTUAL,INSP_LAM0,5,-NA-,1

```

The example shows that the measurement type contains the parameter name. The parameter values are loaded into the Profile_Parameter table.

The orchestration definition file for inspection parameter events has a single orchestration step. For events with the PARAMETERVI event type code and the

value type code ACTUAL, ASSIGN and ASSIGN_DATE calculations are configured, and the profile adapter persists the calculation results to the PROFILE_PARAMETER table.

The next step that allows IBM Predictive Maintenance and Quality to perform the inspection is to store the events that are related to production and inspection. The events for inspection can be in the form of runtime or batch data. Runtime data is raw time series data and batch data is data that is aggregated by day, month, and other units of time. The events are stored in time series tables.

EVENT table

Contains information for master entities that is related to the event, for example, production batch, process, material, and resource.

EVENT_OBSERVATION table

Contains information that is related to the core event, for example, measurement, time of occurrence, and type of event.

Inspection and production events are processed through Predictive Maintenance and Quality Eventload message flows according to the orchestration definition file PMQ_orchestration_definition_inspection.xml.

The orchestration definition file for inspection events has a single orchestration step. For events with the INSPECT and FAIL measurement type code, the TOTAL calculation is configured, and the profile adapter persists the calculation results to the PRODUCT_KPI table.

The orchestration definition file for production events has a single orchestration step. For events with the QTY measurement type code, the TOTAL calculation is configured, and the profile adapter persists the calculation results to the PRODUCT_KPI table.

Event format for inspection loading

Inspection data that consists of production events to report the quantity that is produced, and inspection events to report the inspected and failed products, are loaded as Predictive Maintenance and Quality events.

The classification is based on the event type and the measurement.

For a PRODUCTION event type, the measurement type must be quantity (QTY), and the measurement holds the quantity value.

For an INSPECTION event type, the measurement type must be either INSPECT or FAIL.

- With the INSPECT measurement type, the number of products that underwent inspection is the measurement.
- With the FAIL measurement type, the number of products that failed inspection is the measurement.

The event type and the measurement type must be the key. Other columns that are used are production_batch_code, the location code, the event_start_time, observation_timestamp, and value_type_code. The event_start_time and observation_timestamp indicates the date and time of inspection.

Note: Each PRODUCTION event is followed by two INSPECTION events. Each INSPECTION event has the value 1 and 2 for multirow_no. The INSPECTION events must be in sequence and are not considered as a complete event unless both are included. An INSPECT measurement type must have one more INSPECTION event with FAIL measurement type to complete the action.

Inspection message flow and triggering mechanism

QEWS has two triggering modes, timer-based triggering and file-based triggering.

In the timer-based triggering mode, the QEWS invocation adapter is triggered at the configured time in the batch orchestration file `PMQ_orchestration_definition_batch.xml` once a day.

In the file-based triggering mode, a file with the run date as input is placed in the `batchdatain` directory, and the QEWS invocation adapter is triggered.

The timer-based and file-based trigger flows call the Process Inspection flow, which is shown in the following figure.

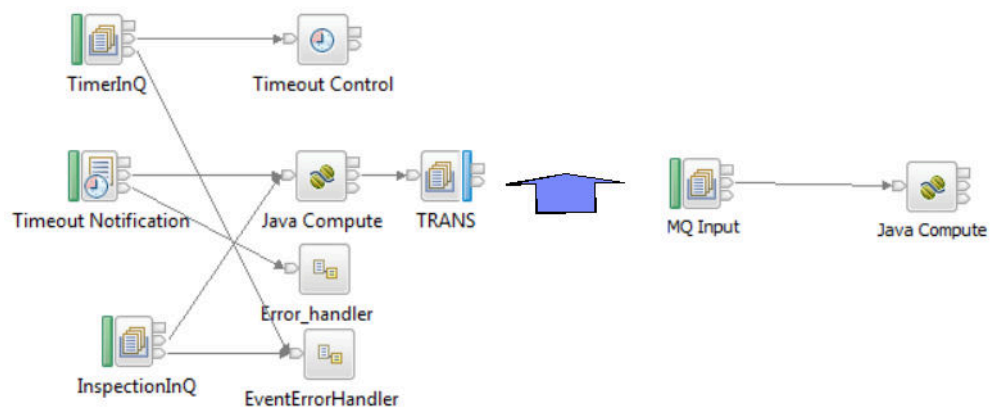


Figure 21. Process Inspection flow

The process inspection flow queries the `PRODUCT_KPI` table, and groups the combinations of masters that are referenced with tenant information that is filtered by the run date. Each combination is passed as a record to the `TRANS` queue.

The `Java Compute` node that is listening to the `TRANS` output queue picks up each combination message and retrieves the records from the `KPI` table, filtering the records by run date, `batch_flag (N)`, and the master combination. The records are then fed to the `QEWS` algorithm.

The invocation adapter also queries the `PROFILE_PARAMETER` table, and loads the parameters for the combination of masters (Product, Process, Material, Location, and Resource). The invocation adapter passes the set of parameters and the inspection data to the `QEWS` algorithm. The `QEWS` algorithm analyzes and persists the data to the `KPI` and `Profile` tables.

Output and reporting

Inspection analysis output is added to the `PRODUCT_KPI` and `PRODUCT_PROFILE` tables. The table structure includes the `Rundate` column. The

historical run analysis outputs in the table remain the same as in previous IBM Predictive Maintenance and Quality releases, and no purging policy is defined. The purging policy might be defined by the business requirements. In addition to the Rundate column, the columns for referring to master entities like Process, Material, Resource, and Location are added to these two tables. These columns are included to group or not group a product, based on reasons such as the material that was used to prepare it, the resource that it uses, and the process that is used to prepare it.

IBM Cognos Business Intelligence includes the Rapidly Adaptive Visualization Engine (RAVE) that is used to build the Failure Rate chart and the Evidence chart. Cognos BI queries the PRODUCT_KPI and PRODUCT_PROFILE tables based on the run date value, and gathers the records that fit the run date value. The records are populated into a .json file at run time, and the .json file is used to prepare the charts.

Results and benefits

The quality early warning system (QEWS) in IBM Predictive Maintenance and Quality reduces cost by detecting problems and issues earlier and more accurately.

Results

Predictive Maintenance and Quality QEWS provides the following results:

- Improves production yields in the manufacturing line.
- Helps you to gain better understanding of root causes of manufacturing issues.
- Provides faster detection of manufacturing quality problems.

Benefits

Subtle changes in failure rates that indicate potential emerging quality problems are detected earlier. Early detection means quicker problem identification, faster problem resolution, and lower total costs.

The definitive nature of QEWS alerts eliminates the need for subjective judgment of statistical process control charts and other traditional tools, providing you with consistent and accurate direction.

QEWS can deliver insightful early warning signals even under variable lot size scenarios.

Warranty

Various conditions can lead to accelerated wear and replacement of manufactured products that are under warranty. Such conditions might include variations in the manufacturing process of the product, variations in the quality of vendors' materials that are used in the product, or the ways in which the product is used.

A small delay in detecting the conditions that lead to accelerated wear can cause more warranty claims and related losses. By understanding the factors that lead to warranty claims, you can take corrective actions such as the following actions:

- Improve manufacturing processes to prevent warranty claims.
- Set pricing for warranties and extended warranties.
- Evaluate vendors of the materials that are used in the product.

The quality early warning system (QEWS) warranty use case in IBM Predictive Maintenance and Quality provides detection that is based on excessive replacement rate and evidence of wear-out.

Replacement rate

QEWS alerts you when the product's random failure rate exceeds a computed threshold. The threshold can reflect product reliability goals (for example, the product population in the field must not exceed a specified failure rate) or financial liability goals (for example, the cost of reimbursing product warranty claims must not exceed a specified total amount).

Wear-out

QEWS alerts you when it finds evidence that product failures are not random in time, but are indicative of wear-out. Wear-out means that products that are in customer use for a longer time fail more often than products that are in customer use for a shorter time. Because wear-out can have serious consequences, QEWS alerts you when it detects evidence of wear-out regardless of how many product units contributed to the detection.

QEWS enables warranty models that are based on sales, production, and manufacturing dates.

Sales model

The Sales model identifies variations in product wear and replacement rates according to the date of sale. The date of sale might correlate with in-service conditions, seasonal climatic conditions, a particular customer, or other important similarities.

For example, a product carries a one-year warranty. In cold conditions, the product becomes brittle and wears prematurely. In certain geographies, products that are sold and enter service in winter initially suffer rapid wear, followed by slower wear during the latter part of the warranty period. The opposite is true for products that are sold and enter service in summer. These seasonal variations affect the product wear rates and weighted replacement rates, which are detected early by QEWS.

Production model

The Production model identifies variations in product wear and replacement rates according to the production date of the product, not the resource in which the product is used. The production date of the product might correlate with the manufacturing equipment operator, the manufacturing process, or other important similarities.

For example, a faulty batch of products is produced during a particular period. The products are installed in resources that have different manufacturing dates. Although the resource manufacturing dates and the product production dates are unrelated, QEWS makes it easier to identify and understand the real cause of the warranty claims.

Manufacturing model

The Manufacturing model identifies variations in product wear and replacement rates according to the manufacturing date of the resource in which the product is used. The resource manufacturing date might correlate with assembly problems that occurred during a particular period.

For example, due to a short-term problem with the manufacturing process of a resource, some of the products that are used in the resource fail

prematurely. Although the resource manufacturing dates and the product production dates are unrelated, QEWS makes it easier to identify and understand the real cause of the warranty claims.

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 data from a network of field service personnel might best be done on a monthly basis.

Business and technical challenges

Fast product cycles, high product volumes, and increasing cost pressure can all lead to increasing numbers of released defective products. The quality early warning system uses IBM technology to detect warranty claim trends earlier than they would otherwise be detected so that you can intervene with corrective action.

Business challenges

Statistical process control methods often overlook cumulative evidence that indicates a worsening quality problem. Better analytical methods are often difficult to implement due to complex computational challenges and constraints in software implementation.

Technical challenges

Premature product wear can have non-obvious causes such as source material variations, seasonal climatic conditions, or temporary manufacturing problems either with the product or with the resource in which the product is used. A small delay in detecting the conditions that lead to accelerated wear can cause more warranty claims and related losses.

Defining the warranty solution

To define the warranty solution, you must load master data, load event data, define message flows, and define the output location of the warranty analysis.

Procedure

1. Load master data. For more information about loading master data, see Chapter 4, "Master data," on page 23.
2. Load event data. You can load event data in batch mode or in real time. For more information about loading event data, see Chapter 5, "Event data," on page 51.
3. Define message flows. For more information about message flows, see "Message flows" on page 15.

Results

IBM Cognos Business Intelligence generates the warranty dashboards and reports.

Warranty solution details

There are requirements that you must consider when loading the master data and event data tables.

Master data tables are loaded by the master flows. The following tables are required to implement a warranty use case:

Master_Location

The Master_Location table contains information that is specific to the geography of the location where the event is produced, or the resource that produces the events.

The following text is an example of a CSV file that is used to load the Master_Location table:

```
location_cd,location_name,region_cd,region_name,country_cd,
country_name,state_province_cd,state_province_name,city_name,
latitude,longitude,language_cd,tenant_cd,is_active
Tokyo,Tokyo,AP,Asia Pacific,JP,Japan,TY,Tokyo,TokyoCity,35.41,139.45,
EN,PMQ,1
```

Master_Resource_Type

The Master_Resource_Type table maintains the Resource type classification. It supports two classification types: ASSET and AGENT. ASSET is a machine or machine part that is used in production. AGENT is the one who operates the machine or the system to ensure that the production procedure is carried out properly.

The following text is an example of a CSV file that is used to load the Master_Resource_Type table:

```
resource_type_cd,resource_type_name,language_cd,tenant_cd
ASSET,Asset,EN,PMQ
AGENT,Agent,EN,PMQ
```

Master_Resource

The Master_Resource table maintains all of the details pertaining to a Resource (ASSET or AGENT). The table maintains information such as which organization hierarchy the resource is lined to, the location at which the resource is installed, the tenant to which the resource is attached or leased, production rate, maintenance interval, and the manufactured date of the resource.

The following text is an example of a CSV file that is used to load the Master_Resource table:

```
resource_cd1,resource_cd2,resource_name,resource_type_cd,
resource_sub_type,parent_resource_cd1,parent_resource_cd2,
standard_production_rate,production_rate_uom,
preventive_maintenance_interval,group_type_cd_1,
group_member_cd_1,group_type_cd_2,group_member_cd_2,
group_type_cd_3,group_member_cd_3,group_type_cd_4,
group_member_cd_4,group_type_cd_5,group_member_cd_5,
location_cd,mfg_date,language_cd,tenant_cd,Is_active
-NA,-NA-,Not Applicable,ASSET,,,,,-NA,-NA,-NA,-NA,-NA-,
-NA,-NA,-NA,-NA,-NA,-TK,2014-01-01,EN,PMQ,1
RCD1,MOD1,RCMOD1,ASSET,,,,,,,,,,,,,TK,,,1
RCD2,MOD2,RCMOD2,ASSET,,,,,-NA,-NA,-NA,-NA,-NA-,
-NA,-NA,-NA,-NA,-NA,-TK,,,1
RCD3,MOD3,RCMOD3,ASSET,,,,,-NA,-NA,-NA,-NA,-NA-,
-NA,-NA,-NA,-NA,-NA,-TK,,,1
```

Master_Product

The Master_Product table contains the core data for the inspection and warranty use cases. This table stores information that is related to a product and the product_type.

The following text is an example of a CSV file that is used to load the Master_Product table:

```

product_cd,product_name,product_type_cd,product_type_name,
language_cd,tenant_cd,Is_active
AAA,TRUNK,B005,Body,EN,PMQ,1
AAB,TRUNK,B005,Body,EN,PMQ,
AAC,TRUNK,B006,Body,EN,PMQ,
AAD,TRUNK,B006,Body,EN,,
AAE,TRUNK,B006,Body,,,

```

Master_Production_Batch

The Master_Production_Batch table contains information about each production batch that is used to produce a product. The information includes the product that is produced, the date it is produced, and the batch information.

The following text is an example of a CSV file that is used to load the Master_Production_Batch table:

```

production_batch_cd,production_batch_name,product_cd,
product_type_cd,produced_date,language_cd,tenant_cd
B1001,FrameBatch,AAA,B005,2012-03-01,EN,PMQ
B1002,FrameBatch,AAB,B005,2012-03-01,EN,PMQ
B1003,FrameBatch,AAC,B006,2012-03-01,EN,PMQ
B1004,FrameBatch,AAA,B006,,,

```

Master_Resource_Production_Batch

The Master_Resource_Production_Batch table contains information about each production batch that is used to produce a resource.

The following text is an example of a CSV file that is used to load the Master_Resource_Production_Batch table:

```

resource_cd1,resource_cd2,production_batch_cd,qty,language_cd
RCD1,MOD1,B005,3,EN
RCD2,MOD2,B006,3,EN
RCD3,MOD3,B005,3,EN

```

Tip:

- If a product can have different parameters (such as LAM0, LAM1, PROB0, CW0, CW1, PROB0), then you can assign a separate product code and production batch to each product variation. Reference each production batch in the Master_Resource_Production_Batch table.
- If a product has the same parameters but different manufacturing or production dates, then you can assign a separate production batch for each manufacturing or production date. Reference each production batch in the Master_Resource_Production_Batch table.

Master data in the Sales model

The following statements apply to the Sales model:

- When a resource is sold, the warranty is tracked from the date of sale until the end of the warranty period. Resources are tracked because, unlike products, in IBM Predictive Maintenance and Quality resources are serialized and can form a hierarchy.
- Each resource contains a number of products. Each product is tracked by a Master_Production_Batch table record.
- The Master_Resource_Production_Batch table handles the mapping between the Master_Resource and Master_Production_Batch tables and also maintains the quantity of products that go into a resource.

Master data in the Production model

The following statements apply to the Production model:

- The warranty for a product extends from the date of production until the end of the warranty period.
- Products are tracked by produced_date.
- The product produced_date is stored in the Master_Production_Batch table and is used as the vintage date.

Master data in the Manufacturing model

The following statements apply to the Manufacturing model:

- The warranty for a resource extends from the date of manufacture until the end of the warranty period.
- Resources are tracked by mfg_date.
- The mfg_date is stored in the Master_Resource table.

Master data and metadata loading

When parameter events are loaded, the event type PARAMETERVW is used. For warranty data events, the event types SALES and WARRANTY are used.

In addition to the measurements that are used in warranty events, measurements with parameter names must also be loaded.

For every measurement type, a unique profile variable is defined so that the Foundation orchestration engine can be leveraged to load parameters to the profile parameter table with multiple master grain levels supported. For parameters loading, profile variables are defined with the profile variable code ParameterName and with the ASSIGN profile calculation configured.

The following figure shows an example of a CSV file that contains the measurement type.

```
1 measurement_type_cd,measurement_type_name,unit_of_measure,carry_forward_indicator,aggregation_type,event
  _code_indicator,language_cd,tenant_cd
2 WARR_LAMO,Warranty Acceptance Level,,0,SUM,0,EN,PMQ
3 WARR_LAML,Warranty Unacceptance Level,,0,SUM,0,EN,PMQ
4 WARR_PROB0,Warranty Confidence Probability Level,,0,SUM,0,EN,PMQ
5 WARR_CW0,Warranty Wearout Acceptance Level,,0,SUM,0,EN,PMQ
6 WARR_CW1,Warranty Wearout Unacceptance Level,,0,SUM,0,EN,PMQ
7 WARR_PROBW0,Warranty Wearout Confidence Probability Level,,0,SUM,0,EN,PMQ
8
```

Figure 22. Example of a CSV file showing the measurement type

The following figure shows an example of a CSV file that contains the parameters for each product.

```

1 profile_variable_cd,profile_variable_name,profile_calculation_name,measurement_type_cd,resource_type_cd,
material_type_cd,profile_units,comparison_string,low_value_date,high_value_date,low_value_number,high_v
lue_number,kpi_indicator,profile_indicator,data_type,aggregation_type,carry_forward_indicator,process_in
dicator,variance_multiplier,language_cd,tenant_cd
2 WARR_LAM0,Warranty Acceptance Level,ASSIGN,WARR_LAM0,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
3 WARR_LAM1,Warranty Unacceptance Level,ASSIGN,WARR_LAM1,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
4 WARR_PROB0,Warranty Probability 0,ASSIGN,WARR_PROB0,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
5 WARR_CW0,Warranty Wearout Acceptance Level,ASSIGN,WARR_CW0,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
6 WARR_CW1,Warranty Wearout Unacceptance Level,ASSIGN,WARR_CW1,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
7 WARR_PROBW0,Warranty Wearout Confidence Level,ASSIGN,WARR_PROBW0,ASSET,-NA,,,,,,,,,1,1,INT,,0,0,1,EN,PMQ
8

```

Figure 23. Example of a CSV file that contains parameters for each product

Event data loading

Warranty uses two types of event data: parameter data and sale and warranty data.

Predictive Maintenance and Quality parameters and their respective combinations covering the master entities Resource, Location, Product, Production Batch, Material, and Process are mapped to Predictive Maintenance and Quality events, and loaded to the PROFILE_PARAMETER table by the orchestration engine using the PMQEventLoad flow, based on the Master_Profile_Variable definition and orchestration definition. Parameter values are stored in the PARAMETER_VALUE column in the PROFILE_PARAMETER table, along with the profile variable and master data references mapped in the event.

The following figure shows an example of a CSV file that is used for loading parameter events.

```

1 incoming_event_cd,event_type_cd,source_system,process_cd,prod_batch_cd,location_cd,event_start_time,even
t_end_time,event_planned_end_time,tenant_cd,operator_cd,model,serial_no,measurement_type_cd,observation
timestamp,value_type_cd,observation_text,measurement_material_code,multirow_no
2 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_LAM0,2014-10-08 00:00:00,ACTUAL,,0.003,,1
3 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_LAM1,2014-10-08
00:00:00,ACTUAL,,0.05722,,1
4 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_PROB0,2014-10-08 00:00:00,ACTUAL,,0.95,,1
5 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_CW0,2014-10-08 00:00:00,ACTUAL,,1,,1
6 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_CW1,2014-10-08 00:00:00,ACTUAL,,1.2,,1
7 1,PARAMETERVW,,,PP9-XX9-009,,2014-10-08 00:00:00,,,PMQ,,,,,WARR_PROBW0,2014-10-08
00:00:00,ACTUAL,,0.99,,1

```

Figure 24. A CSV file that is used for loading parameter events

The orchestration definition XML file for parameter events has a single orchestration step. For events with the event type PARAMETERW and the value type code ACTUAL, the ASSIGN and ASSIGN_DATE calculations are configured, and the Profile adapter persists the calculation results into the PROFILE_PARAMETER table.

After the Master data load flows are completed, you must load the event flows. Event data is loaded on an event basis, where each event is associated with a number of observations. Each observation indicates a measurement type (for example, pressure in kilopascals) and a measurement reading.

The event flows load events such as SALES and WARRANTY that are predefined in the Master_Event_Type table. Every event is related to a particular Resource and the Production_Batch details.

The orchestration definition file for sales and warranty events, PMQ_orchestration_definition_warranty.xml, has a single orchestration step.

Events with the event type SALES and WARRANTY are persisted into the EVENT and EVENT_OBSERVATION tables by the EventStore adapter.

Event data loading in the Sales model

The Sales model event data is loaded in the following order:

1. The SALES event is loaded.
 - The measurement_type_cd field contains SALESDATE.
 - The event_start_time field and the observation_timestamp field contain the date of sale.
 - The observation_text field contains the warranty end date. By default, the value is three years but it can be changed as required.
 - The measurement field contains the number of warranty months.
2. Any number of WARRANTY events are loaded.
 - The measurement_type_cd field contains WARRANTYINDICATOR.
 - The event_start_time field and the observation_timestamp field contain the date when the claim was made.
 - The observation_text field and the measurement field are blank.

The following text is an example of a CSV file that is used for loading sales events.

```
incoming_event_cd,event_type_cd,source_system,process_cd,
prod_batch_cd,location_cd,event_start_time,event_end_time,
event_planned_end_time,tenant_cd,operator_cd,resource_cd2,
resource_cd1,measurement_type_cd,observation_timestamp,
value_type_cd,observation_text,measurement,material_code,multirow_no
1,SALES,,B1001,Tokyo,2006-12-19T12:00:00,,PMQ,,MOD1,RCD1,
SALESDATE,2006-12-19T12:00:00,ACTUAL,12/19/2009,35.9344262295082,,1
1,WARRANTY,,B1001,Tokyo,2013-06-17T12:00:00,,PMQ,,MOD1,RCD1,
WARRANTYINDICATOR,2013-06-17T12:00:00,ACTUAL,N,,1
1,SALES,,B1002,Tokyo,2006-11-20T12:00:00,,PMQ,,MOD2,RCD2,
SALESDATE,2006-11-20T12:00:00,ACTUAL,11/20/2009,35.9344262295082,,1
1,WARRANTY,,B1002,Tokyo,2009-05-04T12:00:00,,PMQ,,MOD2,RCD2,
WARRANTYINDICATOR,2009-05-04T12:00:00,ACTUAL,Y,,1
1,SALES,,B1003,Tokyo,2006-10-31T12:00:00,,PMQ,,MOD3,RCD3,
SALESDATE,2006-10-31T12:00:00,ACTUAL,10/31/2009,35.9344262295082,,1
```

Event data loading in the Production model

The Production model event data is loaded in the following order:

1. The SALES event is loaded.
 - The measurement_type_cd field contains SALESDATE.
 - The event_start_time field and the observation_timestamp field contain the Produced Date from the Master_Production_Batch table.
 - The observation_text field contains the warranty end date. By default, the value is 3 years but it can be changed as required.
 - The measurement field contains the number of warranty months.
2. Any number of WARRANTY events are loaded.
 - The measurement_type_cd field contains WARRANTYINDICATOR.
 - The event_start_time field and the observation_timestamp field contain the date when the claim was made.
 - The observation_text field and the measurement field are blank.

Event data loading in the Manufacturing model

The Manufacturing model event data is loaded in the following order:

1. The SALES event is loaded.
 - The measurement_type_cd field contains SALESDATE.
 - The event_start_time field and the observation_timestamp field contain mfg_date from the Master_Resource table.
 - The observation_text field contains the warranty end date. By default, the value is 3 years but it can be changed as required.
 - The measurement field contains the number of warranty months.
2. Any number of WARRANTY events are loaded.
 - The measurement_type_cd field contains WARRANTYINDICATOR.
 - The event_start_time field and the observation_timestamp field contain the date when the claim was made.
 - The observation_text field and the measurement field are blank.

Warranty triggering

There are two warranty triggering modes, timer-based triggering and file-based triggering.

In the timer-based triggering mode, the warranty IBM SPSS job is triggered at the scheduled time once a day, as configured in the batch orchestration file `PMQ_orchestration_definition_batch.xml`, with the current date as the run date. The default sub use case is Sales.

The Batch integration AutoTrigger flow accepts input parameters such as the sub use case name from the scheduler configuration (in the XML file), in addition to the schedule time, the queue name, and the duration. The AutoTrigger flow places the timer request to the queue that is configured to trigger the SPSSJobIntegration flow, which in turn triggers the SPSS job at the scheduled time, using the configurations and parameters defined in the batch orchestration.

In the file-based triggering mode, the warranty SPSS job is triggered by placing the run date file in the `batchdatain` directory with the run date and the sub use case parameter. The WarrantyDataPreparation flow accepts the run date file, and places an MQ request in the `PMQ.JOBINTEGRATION.IN` queue to trigger the SPSSJobIntegration flow, which in turn triggers the SPSS job using the configurations and parameters defined in the batch orchestration.

The following figure shows the batch orchestration configuration for warranty. Configurations can be modified at run time.

orchestration	
Identifier	SALES
scheduler	
scheduled_time	00:00:00
queue_name	PMQ.QEWS.WSTIMER.IN
duration_in_days	1
webservice	Webservice configuration for Warranty SALES
url	http://9.122.126.168:9080/process/services/ProcessManagement
jobLocationURI	spssc:///?id=5691007b90f455850000014a28e6e3bc939a
parameters	
parameter	
name	RunDateInFormatYYYYMMDDHyphenSeparated
value	StartDate
type	dynamic
parameter	
name	ServiceTabQtyMultiplier
value	1
type	static
parameter	
name	IsRunDateEqServerDate
value	0
type	static
parameter	
name	SubUseCase
value	SALES
type	static
notificationEnabled	true

Figure 25. Batch configuration for the warranty use case

Data in the event and event observation tables must be processed so that it can be provided to QEWS. Processing the tables involves calling the SPSS modeler stream, which picks data from Event, Event_Observation, Resource, Product, and Production_Batch and prepares the data in the following format:

```
Product_id | Produced Date | Service_Month | Parts under Warranty | Parts
replaced | tenant_cd | Use Case | RunDate
```

A Service table holds these records and forms as input from where the QEWSL invocation algorithm picks the data.

Once the SPSS modeler stream completes the transformation of Masters and Events into Service details, the QEWS invocation flow is triggered.

SPSS sends a status file with the run date to the integrationin directory of the ESB node, and the file is processed by the WarrantyFileIntegration flow. When the status of the warranty job is SUCCESS, it triggers the ProcessWarranty flow by placing a message in the input queue of ProcessWarranty.

The ProcessWarranty flow consumes the run date from the Status message, and queries the Service table in the Predictive Maintenance and Quality data mart. The flow prepares object structures comprised of the Parts under warranty (WPARTS), Parts replaced under warranty (WREPL), Produced Date, product_id, and the Product Parameters.

SPSS Modeler Streams

There are two SPSS Modeler streams and corresponding Collaboration and Deployment Services jobs for Warranty. The first stream is for the Manufacturing and Production models, in which the specific use case can be controlled by toggling across a parameter from MFG (Manufacturing) to PROD (Production). The second stream is for the Sales model.

The streams differ in the transformation logic for producing the Service table (for more information, see “Service tables”). The SPSS Modeling layer provides special logic for each of the models; all other processing and treatment is the same for all of the models.

The main difference between the models is in the aggregation and tracking of vintages. A vintage is a combination of the product ID (numbered product type) and a date (date of sale, date of production, or date of manufacture). The date at which the product was put in service is assumed to be the same as the date of sale of the resource in which the product is used. The models do take into account the differential tracking and treatment of products that are sold or shipped as replacements of other products that were shipped separately. Replacement products can either be excluded from the event structure or they can be included as a separate vintage.

You can choose between the Production and Manufacturing models by changing the `IsMFG_OR_PROD` variable of the `IBM_QEWSL_JOB` C&DS job to `PROD` or `MFG`. You can change the variable from either SPSS Collaboration and Deployment Services (during ad-hoc one time trigger) or IIB (during automated triggers).

The Sales model is controlled by a separate job named `IBMMPMQ_QEWSL_SALES_JOB`. The job can be run from IIB by using its job URI.

Customizable Parameters and Special Scenarios

Both SPSS Modeler streams contain some common parameters that can be used while running the SPSS models under special scenarios and requirements. These options can be altered from the SPSS Collaboration and Deployment Services Job Variable or from IIB. The preferred way of altering these parameters is through IIB. The description and uses of these parameters is as follows:

IsRunDateEqServerDate

This parameter determines whether the SPSS server system date (value = 1) or a custom run date (value = 0) is used in computation logic requiring a run date. The default value is 0 and it uses the custom run date that is supplied by IIB (corresponding to the IIB server system date during default runs).

RunDateInFormatYYYYMMDDHyphenSeparated

This parameter is used only if the value of `IsRunDateEqServerDate` parameter is 0. The parameter sets the custom run date. The required date format is `YYYY-MM-DD`.

ServiceTabQtyMultiplier

For performance reasons, sometimes you might need to run the `QEWSL` warranty engine on a sample of the complete data. `QEWSL` is a weighted algorithm, so by default it does not produce the same graphs or alerts for a sample as it would for the complete data. If the sample is a good true representative sample, this parameter helps to correct the scale of the weighted results or plots to give a representative output. The parameter is set with a value of multiplier as $1/number$.

Service tables

When the SPSS stream runs, it populates a DB2® table named `PMQSCH.SERVICE` (referred to as the Service table). After the table is populated, the processing is the same for all models.

The structure of the Service table is the same for all models. What changes is the computation and aggregation logic for the table fields by the different SPSS streams and models.

The Service table contains the following fields:

PRODUCED_DATE

This field contains the vintage date of the Sales or Manufacturing model. Together with the PRODUCT_ID field, this field represents the vintage of the record. Together with PRODUCT_ID and SVC_MNTHS fields, this field represents the composite unique key for the table.

PRODUCT_ID

This field represents the non-serialized product identifier (numeric product type) for the product whose replacement needs to be tracked.

SVC_MNTHS

This field represents the number of months that any of the products of that vintage (PRODUCED_DATE + PRODUCT_ID) were in service during their warranty period. For example, a three-year warranty period can contain up to 36 service months.

To have a consistent number of maximum service months across vintages in a computation lot, products with shorter warranty periods (for example, two years) can be given more SVC_MNTHS to match products with longer warranty periods (for example, 36 months). In this case, during the SVC_MNTHS that are outside of the warranty period, WPARTS and WREPL are both 0.

WPARTS

This field represents the number of products of that vintage (PRODUCED_DATE + PRODUCT_ID) that were in service without any warranty claims during the service month (SVC_MNTHS).

WREPL

This field represents the number of products of that vintage (PRODUCED_DATE + PRODUCT_ID) that failed (received a warranty claim) during the service month (SVC_MNTHS).

TENANT_ID

This field is an identifier to differentiate between tenant data in a multi-tenant environment.

Warranty message flow and triggering mechanism

When the SPSS modeler stream runs successfully, it invokes the warranty flow. A status message that is embedded with a date value is placed in the PMQ.QEWS.WARRANTY.IN queue. When the broker interface detects a message in the queue, it triggers the QEWSL algorithm. The embedded date value in the message is the rundate, which becomes the date of reference for the warranty flow. The Service table records and the parameters are passed to the QEWSL algorithm.

The same message flow is used to trigger all of the warranty models.

Output and reporting

The output from the algorithm is persisted in the Lifetime_KPI and Lifetime_Profile tables. In these tables, along with the analysis output, there are columns for Rundate and UseCase. In addition, columns that refer to the master

entities Resource, Process, Material, and Location are added to these two tables. Currently, the references to the master entities are not used. They refer to the NA rows for the default language.

The Replacement rate chart and the evidence charts are prepared by the Rapidly Adaptive Visualization Engine (RAVE) in IBM Cognos Business Intelligence, and they extract the records from the Lifetime KPI and Profile tables for a given rundate and use case.

Results and benefits

The quality early warning system (QEWS) warranty use case in IBM Predictive Maintenance and Quality reduces cost by detecting problems and issues earlier than they would otherwise be detected, and more accurately.

Results

IBM Predictive Maintenance and Quality QEWS delivers the following results:

- Shows you where to improve manufacturing processes to prevent warranty claims.
- Helps you to set pricing for warranties and extended warranties.
- Helps you to evaluate vendors of the materials that are used in products.

Benefits

Subtle changes in warranty claim rates indicative of potential emerging quality problems are detected earlier. This allows for quicker problem identification, faster problem resolution, and lower total costs.

The definitive nature of QEWS alerts eliminates the need for subjective judgment of statistical process control charts and other traditional tools, providing you with consistent and accurate direction.

QEWS can deliver insightful early warning signals even under variable lot size scenarios.

Parametric

The quality early warning system (QEWS) parametric use case in IBM Predictive Maintenance and Quality detects unfavorable changes in variable-type data, and provides information that facilitates diagnostics and alarm prioritization.

The QEWS parametric use case uses the Quality Early Warning System for Variable Data (QEWSV) algorithm to monitor variable-type data. This type of data is found in a number of 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.

Predictive Maintenance and Quality maintains variable data in the event store, and prepares data for the QEWSV algorithm. Variable values and evidence charts are plotted by using parametric results.

The QEWSV system is organized around three basic notions that are referred to as Variables, Operations, and Time slides. The behavior of selected variables is monitored based on acceptable and unacceptable characteristics of the underlying

process. These characteristics are converted to rules of the decision-making scheme that is used to decide whether a variable is flagged. Operations refer to points in the process that have a potential of influencing the stochastic behavior of the variables and are thus considered as likely problem areas when variables show unacceptable behavior. Time slides are data structures that organize measurements that pertain to a particular variable regarding an operation of interest.

Mapping

In Predictive Maintenance and Quality, Operation is synonymous to Process. Operation is the sequence of flow that is involved in arriving at the end product or intermediate product. Additionally, a tool is also considered a factor in identifying the behavior. In Predictive Maintenance and Quality, the tool is considered the Resource.

Time slides deal with the interval of time on which the measurement or observation on a variable is taken. The observation time stamp is more synonymous with time slides when an event observation is made.

Variables are defined for every operation per tool. In Predictive Maintenance and Quality, Variables are equated with Measurement type, whose measurements are read at different time intervals during the sequence of the operation flow.

Analysis performed in parametric use case

Based on observations with a specific measurement type along the time slides, the deviation or drift from targeted values are computed and analyzed to show whether the sequence of process is adhering to normal operation limits. This analysis impacts the quality of the end or intermediate product.

Sub use cases

Predictive Maintenance and Quality can handle a varied set of master data sets, from end products to manufacturing machinery to the raw materials used, as well as the environment or location-specific data. Predictive Maintenance and Quality identifies the following categories of analysis that is done on each of the master entities. Some categories might be a combination of different masters or a lone entity.

Process resource validation

This category is the default use case, where the process and the resource that takes part in the process is monitored based on a defined set of variables. These variables are associated with a set of parameters that define the target values, acceptable limit, unacceptable limit, standard deviation, false alarm rate, and unacceptable factor.

Resource validation

A resource is monitored based on the standard operation limits across a few measurement types (variables). This type of health check is essential in identifying any issues in the resource and correcting those issues to improve the performance and throughput.

Product validation

With Quality inspection, the product as a whole is checked, based on the failure rate. In variable data, given the set of variables whose targets are set for the product to meet, any deviation or drift beyond the allowed deviation highlights a flaw in the product.

Material validation

Raw materials purchased from a vendor are monitored for a defined set of guidelines as variables, and validated to check on the quality of the procured material.

Location suitability

With variable analysis, a location is analyzed to see whether it is suitable for a particular operation. Variables like pressure, temperature, humidity, and their time slide values can forecast the suitability of a location for carrying out any operation.

Each validation supports grains of Resource, Process, Material, Product, and Location. Given a combination of grains, a variable can be declared and provided with a target set of parameters to validate.

Business and technical challenges

The parametric use case has business and technical challenges.

Business challenges

The business challenges lie in identifying the rules to set up the quality norms for an end product or a raw material. When a rule is defined, if it does not identify a flaw in a product or material, it leads to noise and turbulence in the confidence of the quality. When there are more quality-related issues, there is more damage to the business, and the cost that is incurred in replacements and servicing is greater.

Applying complex statistical computation is a difficult task, and hard to implement with market available packages.

Technical challenges

Identifying a quality defect is difficult, unless the defect is treated under different variable conditions. Usually, in the quality inspection process, the quality is defined by a set of rules. However, defining those rules cannot identify subtle variations that occur during the manufacturing process. Therefore, treating defects under various conditional checks across different measurements, based on a target value, helps in forecasting the type of flaw that might cause a product to fail. Implementing conditional checks is a difficult task, as it requires complex statistical procedures.

Defining the parametric solution

To define the parametric solution, you must load master data, load event data, define message flows, and define the output location of the parametric analysis.

Procedure

1. Load master data. The master data includes master records for Process, Resource, Product, Material, and Location. For more information about loading master data, see Chapter 4, "Master data," on page 23.
2. Load metadata. Metadata includes Measurement type (Variable), Event type, and Profile variable data.
3. Load event data. You can load event data in batch mode or in real time. Event data includes parameter data and measurement observations at every time scale for each defined event type. For more information about loading event data, see Chapter 5, "Event data," on page 51.

Results

IBM Cognos Business Intelligence uses data in the PARAMETRIC_KPI and PARAMETRIC_PROFILE tables to generate the parametric dashboard and reports.

Parametric solution details

There are requirements that you must consider when you load the master data, metadata, and event data.

Master data and metadata loading

Master data loading involves the loading of all master entities, such as Process, Resource, Product, Material, and Location.

Separate from the entities, Measurement type that is specific to a resource, product, process, or location must be loaded. For example, if temperature (TEMP) is the variable that is monitored, then the measurement type code for resource validation can be TEMP_R, and the measurement type code for location suitability can be TEMP_L.

In addition to the Measurement type, Event types must be loaded. An event type is dedicated to each sub use case. The following table describes the mapping of sub use cases to event type.

Table 16. Sub use case to event type mapping

Sub use case	Event type
Process - Resource Validation	PRVARIABLE
Resource Validation	RVARIABLE
Product Validation	PVARIABLE
Material Validation	MVARIABLE
Location Suitability	LVARIABLE

Each sub use case is identified by the defined set of Event types. When a sub use case is triggered, then data for the event type is fetched from the Event and Event Observation tables.

For every measurement type, a unique profile variable is defined so that the Analytics Solutions Foundation orchestration engine is leveraged to load parameters to profile parameter tables, with multiple master grain levels supported. When parameters are loaded, profile variables are defined with the profile variable code *MeasurementTypeCd_ParameterName*, where *MeasurementTypeCd* is the variable code with the PROFILE_PARAMETER_ASSIGN calculation configured.

Before you can load parameters into tables, you must define measurement type entries. For example, to measure thickness, this measurement type must be loaded. To begin, you load the Measurement type table. The following text is an example of loading the Measurement type table.

```
measurement_type_cd,measurement_type_name,unit_of_measure,carry_forward_indicator,
aggregation_type,event_code_indicator,language_cd,tenant_cd
THICKNESS_P,Thickness,,0,SUM,0,EN,PMQ
```

When the Measurement type table is loaded, you must load the Master_Profile_Variable table. The parameter name has values such as THICKNESS_P_TARGET, THICKNESS_P_SIGMA, THICKNESS_P_LAM0, THICKNESS_P_LAM1, THICKNESS_P_CONTROL, THICKNESS_P_FALSEALARMRATE, THICKNESS_P_UNACCEPTFACTORSIGMA, and THICKNESS_P_NO_DAYS.

The following CSV file is an example of the Master_Profile_Variable table.

```
profile_variable_cd,profile_variable_name,profile_calculation_name,
measurement_type_cd,resource_type_cd,material_type_cd,profile_units,comparison_string,
low_value_date,high_value_date,low_value_number,high_value_number,kpi_indicator,
profile_indicator,data_type,aggregation_type,carry_forward_indicator,process_indicator,
variance_multiplier,language_cd,tenant_cd
THICKNESS_P_TARGET,Thickness Target,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,1,
EN,PMQ
THICKNESS_P_SIGMA,Thickness Sigma,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,1,
EN,PMQ
THICKNESS_P_LAM0,Thickness Lam0,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,1,
EN,PMQ
THICKNESS_P_LAM1,Thickness Lam1,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,1,
EN,PMQ
THICKNESS_P_CONTROL,Thickness Control,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,1,
EN,PMQ
THICKNESS_P_FALSEALARMRATE,Thickness FalseAlarmRate,ASSIGN,THICKNESS_P,ASSET,
-NA-,,,,,,1,1,INT,,0,0,1,EN,PMQ
THICKNESS_P_UNACCEPTFACTORSIGMA,Thickness UacceptFactorSigma,ASSIGN,THICKNESS_P,ASSET,-
NA-,,,,,,1,1,INT,,0,0,1,EN,PMQ
THICKNESS_P_NO_DAYS,Thickness no of days,ASSIGN,THICKNESS_P,ASSET,-NA-,,,,,,1,1,INT,,0,0,
1,EN,PMQ
```

Note that the profile variable connects the measurement type to the parameter name.

Event data loading

Parametric data uses the following two types of event data.

parameter data

Parameters remain unique for a combination of master entities. Parameters are loaded to the PROFILE_PARAMETER table.

variable data

Variable and observation reading data are loaded to the Event and Event Observation tables, supporting different master data grains.

If parameter and variable data must be linked to process, product, resource, location, material, or a combination of these entities, appropriate grains in the event message must be set.

Parameter event loading

Before you load parameters, you must understand the parameters that are required for the parametric use case. The following table describes the parameters that Predictive Maintenance and Quality uses.

Table 17. Parametric parameters

PMQ parameter	QEWS parameter	Description
SIGMA	Sigma	The assumed standard deviation of the measurements. This value must always be greater than zero.

Table 17. Parametric parameters (continued)

PMQ parameter	QEWS parameter	Description
TARGET	Target	The most desirable value for the center of the measurement population. Typically interpreted as the best level for the mean of the measurements.
LAM0	Accept_Level	The level of mean of the measurements that is still considered acceptable. Typically, this level is close to the Target, and it reflects the amount of wiggle room that remains for the population mean around the target. In many cases involving low process capability, this level coincides with the Target, indicating that there is no wiggle room for the population mean.
LAM1	Unaccept_Level	The level of mean of the measurements that is considered unacceptable. This is the level for which good detection capability is desired. Generally, an unacceptable level should be further away from the Target than the acceptable level. It is also advisable to keep a certain degree of separation between the acceptable and unacceptable levels (for example, no lower than $0.2 * \text{Sigma}$, where possible).
CONTROL	Type_of_Control	1 means that the control is one-sided (you are only interested in detecting changes up or down). 2 means that the control is two-sided: both types of deviation from the Target are considered unacceptable. If Type_of_Control = 1 and Accept_Level < Unaccept_Level, then only changes of the process mean up are of interest. If Type_of_Control = 1 and Accept_Level > Unaccept_Level, then only changes of the process mean down are of interest. If Type_of_Control = 2, then you can 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.
FALSEALARM RATE	False_Alarm_Rate	Default = 1000, which means that the detection procedure produces a rate of false alarms of 1 per 1000 points (that is, 1 per 1000 values in the *.tsd data file) when the population mean is located at the Accept_Level.
UNACCEPTFACTOR SIGMA	Unaccept_Factor_Sigma	Presently not used.

Parameters, and combinations of parameters for the Resource, Location, Product, Material, and Process master entities are mapped to IBM Predictive Maintenance and Quality events. By using the PMQEventLoad flow in the orchestration engine, the parameters are loaded to the PROFILE_PARAMETER table, based on the Master_Profile_Variable definition and the orchestration definition. Parameter values are stored in the PARAMETER_VALUE column of the PROFILE_PARAMETER table, along with the profile variable and master data references mapped in the event.

When parameters are loaded, profile variables are defined with the profile variable code *MeasurementTypeCd_ParameterName*, where *MeasurementTypeCd* is the variable code. The PROFILE_PARAMETER_ASSIGN profile calculation is used to load parameters to the PROFILE_PARAMETER table.

Parameter event mapping

The following table describes the mapping of parameters to events.

Table 18. Parameter to event mapping

Parameter	Predictive Maintenance and Quality event
incoming_event_cd	incoming_event_cd
Hard coded to PARAMETERV	event_type_cd
	source_system_cd
process_cd (if applicable)	process_cd
production_batch_cd (if applicable)	production_batch_cd
	location_cd
Parameter load time/Parameter update time	event_start_time
	event_end_time
	event_planned_end_time
tenant_cd	tenant_cd
	operator_cd
Model (if applicable)	model
serial_no (if applicable)	serial_no
Variable code	measurement_type_cd
Parameter load time/Parameter update time	observation_timestamp
Hard coded to ACTUAL	value_type_cd
ProfileVariableCd (<i>MeasurementTypeCd_ParameterName</i>)	observation_text
Parameter value	measurement
material_cd (if applicable)	material_cd
multirow_no	multirow_no

Parameter events processing

Parameter events are processed through Predictive Maintenance and Quality Eventload message flows according to the orchestration definition file.

The orchestration definition file for parameter events is named *PMQ_orchestration_definition_parameter.xml*, and it has a single orchestration step. For events with the PARAMETERV event type code and the ACTUAL value type, the PROFILE_PARAMETER_ASSIGN calculation is configured, and the profile adapter adds parameters to PROFILE_PARAMETER tables.

The following text is an example of loading parameter event data.

```
incoming_event_cd,event_type_cd,source_system_cd,process_cd,prod_batch_cd,
location_cd,event_start_time,event_end_time,event_planned_end_time,tenant_cd,
operator_cd,model,serialNo,measurement_type_cd,observation_timestamp,value_type_cd,
observation_text,measurement,material_code,multirow_no
```

```

1,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:00,2014-11-26 00:00:00,
2014-11-26 00:00:00,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:00,ACTUAL,
THICKNESS_P_FALSEALARMRATE,1000,-NA-,1
2,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:01,2014-11-26 00:00:01,
2014-11-26 00:00:01,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:01,ACTUAL,
THICKNESS_P_LAM0,0.85,-NA-,1
3,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:02,2014-11-26 00:00:02,
2014-11-26 00:00:02,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:02,ACTUAL,
THICKNESS_P_LAM1,0.9,-NA-,1
4,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:03,2014-11-26 00:00:03,
2014-11-26 00:00:03,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:03,ACTUAL,
THICKNESS_P_CONTROL,2,-NA-,15,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26
00:00:04,2014-11-26 00:00:04,2014-11-26 00:00:04,PMQ,,-NA-,-NA-,THICKNESS_P,
2014-11-26 00:00:04,ACTUAL,THICKNESS_P_SIGMA,0.04,-NA-,1
6,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:05,2014-11-26 00:00:05,
2014-11-26 00:00:05,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:05,ACTUAL,
THICKNESS_P_TARGET,0.8,-NA-,1
7,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:06,2014-11-26 00:00:06,
2014-11-26 00:00:06,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:06,ACTUAL,
THICKNESS_P_UACCEPTFACTORSIGMA,1.5,-NA-,1
8,PARAMETERV,,-NA-,PP9-XX9-009,-NA-,2014-11-26 00:00:07,2014-11-26 00:00:07,
2014-11-26 00:00:07,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-26 00:00:07,ACTUAL,
THICKNESS_P_NO_DAYS,2000,-NA-,1

```

Variable event loading

Variable data is made up of the measurements and observation readings that are taken at different time intervals during the sequence of the operation flow. Observation readings or variable data are mapped to Predictive Maintenance and Quality events, and loaded to event and event observation tables. The orchestration engine is used to load the observation readings or variable data by using the PMQEventLoad flow, based on the orchestration definition. Master data references in the event vary, depending on the type of event and sub use case.

Variable event mapping

The following table describes the parameter to event mapping for variables.

Table 19. Parameter to event mapping for variables

Parameter	Predictive Maintenance and Quality event
incoming_event_cd	incoming_event_cd
Hard coded to event type, based on sub use case chosen (PRVARIABLE, RVARIABLE, PVARIABLE, MVARIABLE, LVARIABLE)	event_type_cd
	source_system_cd
process_cd (if applicable)	process_cd
production_batch_cd (if applicable)	production_batch_cd
location_cd (if applicable)	location_cd
event_start_time	event_start_time
	event_end_time
	event_planned_end_time
tenant_cd	tenant_cd
	operator_cd
Model (if applicable)	model
serial_no (if applicable)	serial_no

Table 19. Parameter to event mapping for variables (continued)

Parameter	Predictive Maintenance and Quality event
Variable code	measurement_type_cd
observation_timestamp	observation_timestamp
Hard coded to ACTUAL	value_type_cd
	observation_text
Observation reading or variable value	measurement
material_cd (if applicable)	material_cd
multirow_no	multirow_no

Master data references in the event vary, depending on the type of event or sub use case.

The following text is an example of variable event data.

```
incoming_event_cd,event_type_cd,source_system_cd,process_cd,prod_batch_cd,location_cd,
event_start_time,event_end_time,eventlanned_end_time,tenant_cd,operator_cd,model,serialNo,
measurement_type_cd,observation_timestamp,value_type_cd,observation_text,measurement,
material_code,multirow_no
1,PBVARIABLE,,-NA-,PP9-XX9-009,-NA-,2014-11-28 01:10:59,2014-11-28 01:10:59,
2014-11-28 01:10:59,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-28 01:10:59,ACTUAL,,0.75,-NA-,1
2,PBVARIABLE,,-NA-,PP9-XX9-009,-NA-,2014-11-28 02:10:59,2014-11-28 02:10:59,
2014-11-28 02:10:59,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-28 02:10:59,ACTUAL,,0.79,-NA-,1
3,PBVARIABLE,,-NA-,PP9-XX9-009,-NA-,2014-11-28 03:10:59,2014-11-28 03:10:59,
2014-11-28 03:10:59,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-28 03:10:59,ACTUAL,,0.79,-NA-,1
4,PBVARIABLE,,-NA-,PP9-XX9-009,-NA-,2014-11-28 04:10:59,2014-11-28 04:10:59,
2014-11-28 04:10:59,PMQ,,-NA-,-NA-,THICKNESS_P,2014-11-28 04:10:59,ACTUAL,,0.77,-NA-,1
```

Variable events processing

Variable events are processed through Predictive Maintenance and Quality Eventload message flows according to the orchestration definition file.

The orchestration definition file for parametric events is named `PMQ_orchestration_definition_parametric.xml`, and it has a single orchestration step. The event store adapter stores variable raw events into event and event observation tables. The event type is used to differentiate between the events of different sub use cases.

Parametric orchestration and triggering mechanism

The parametric use case has the following batch triggering modes:

- Timer-based triggering, which passes the run date as the current date, and passes the sub use case name to the algorithm invocation flow.
- File-based triggering, which passes the run date and the event type code (the sub use case) as input.

Depending on the parametric use case, the `SubUseCase` value is configured:

- RVALIDATION for resource health check
- PRVALIDATION for process resource validation
- MVALIDATION for material validation
- PBVALIDATION for product validation
- LVALIDATION for location suitability

Timer-based triggering

The orchestration definition file `PMQ_orchestration_definition_batch.xml`, in the `properties` directory on the Integration Bus node computer, is configured to trigger the parametric timer at the configured scheduled time once per day. The `AutoTrigger` flow of the batch integration process flow accepts an input parameter, such as the `SubUseCase` name, from the scheduler configuration, in addition to the schedule time, the queue name, and the duration. The `AutoTrigger` flow places the timer request to the configured queue, which triggers the `ProcessParametric` flow at the scheduled time. The `ProcessParametric` uses the `SubUseCase` name and the run date (as the current date) as inputs. The `ProcessParametric` flow invokes the `QEWS` algorithm.

You can change the `SubUseCase` name and timer configuration in the orchestration definition file at run time.

File-based triggering

In file-based triggering, a file with the naming convention `parametric_rundate*.txt` must be placed in the `batchdatain` folder. The contents of the file must have the following format:

```
rundate=2014-12-01  
subusecase=PBVALIDATION
```

The file is picked up by the `ParametricDataPreparation` flow, which is included in the `PMQQEWSIntegration` application. The `ParametricDataPreparation` flow converts the file into an WebSphere MQ message, and places the message in the `PMQ.QEWS.PARAMETRIC.IN` queue. The `ProcessParametric` flow is triggered, with the `SubUseCase` name and the run date as inputs.

Parametric algorithm invocation

The following diagram shows how the parametric algorithm is invoked.

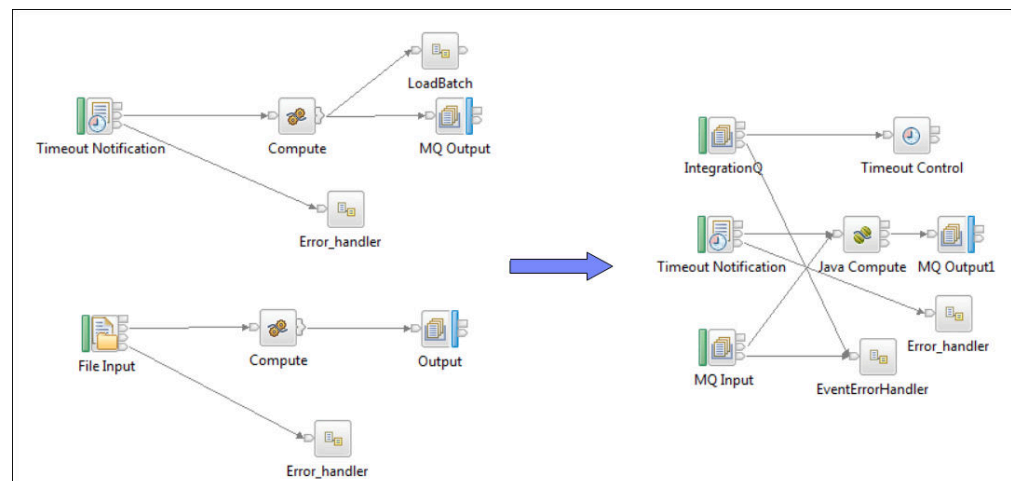


Figure 26. Parametric algorithm invocation

Output and reporting

After the parametric algorithm is invoked by using the timer-based or file-based triggering mode, the `Event` and `Event Observation` tables are queried for a specific

event type and date range. The date range covers a specified number of days backward from the run date. Each sub use case has a specific event type.

The specified number of days is taken from the NO_DAYS parameter. When the grain combination is identified, the parameters are retrieved from the PROFILE_PARAMETER table.

The data set from the Event and Event Observation tables and the parameters are passed to the parametric algorithm. The results of the parametric analysis are persisted into the PARAMETRIC_KPI and PARAMETRIC_PROFILE tables. These two tables include the Rundate and Event type ID columns. The Rundate column holds the date when the QWESV algorithm was triggered and when the execution began. The Event Type ID column refers to the Master Event Type table, where the Event type code for each of the sub use cases Process, Resource Validation, Material Validation, Location Validation, and Product Validation is stored.

The Rundate column provides the option of storing the processed data based on the execution date, which maintains the processed records of earlier runs. Currently, there is no purging policy set for the KPI and Profile tables. The purging policy may later be defined based on the business requirements.

IBM Cognos Business Intelligence includes the Rapidly Adaptive Visualization Engine (RAVE) that is used to build the Parametric chart. CognosBI queries the PARAMETRIC_KPI and PARAMETRIC_PROFILE tables based on the run date value, and gathers the records that fit the run date value. The records are populated into a .json file at run time, and the .json file is used to prepare the charts.

Results and benefits

The quality early warning system for variables (QEWSV) in IBM Predictive Maintenance and Quality reduces cost by detecting problems and issues earlier and more accurately.

Results

Predictive Maintenance and Quality QEWSV delivers the following results:

- Improves production yields in the manufacturing line.
- Helps you to gain a better understanding of root causes of manufacturing issues by dealing directly with Variables and Operations.
- Provides faster detection of manufacturing quality problems.

Benefits

Subtle changes in failure rates that indicate potential emerging quality problems are detected earlier. Early detection means quicker problem identification, faster problem resolution, and lower total costs. The definitive nature of QEWSV alerts eliminates the need for subjective judgment of statistical process control charts and other traditional tools, providing you with consistent and accurate direction. QEWSV can deliver insightful early warning signals, even under variable lot size scenarios.

Chapter 7. IBM Insights Foundation for Energy

The IBM Predictive Maintenance and Quality solution provides health scores of assets that are commissioned in IBM Insights Foundation for Energy.

IBM Insights Foundation for Energy is a cloud-based data management, visualization, and analytics solution that can help energy providers avoid asset failures, improve asset utilization, optimize network availability, and decrease loss of service. Predictive Maintenance and Quality provides health scores based on the raw input data and on features that are computed in models.

Predictive Maintenance and Quality provides the following message flows to load Insights Foundation for Energy data and obtain the predicted results for the asset:

Commission metadata

Predictive Maintenance and Quality loads measurement types for various asset classes.

Commission master data

Predictive Maintenance and Quality loads asset master data such as serial number, and asset static properties, such as `installationDate` and `removalDate`.

Load measurement data

Predictive Maintenance and Quality loads reading data for an asset.

Query predicted results

Predictive Maintenance and Quality uses a Representational State Transfer (REST) service to obtain the predicted results for an asset class.

Commission metadata and master data

Predictive Maintenance and Quality uses the following process to commission metadata and master data.

1. Insights Foundation for Energy sends metadata and master data of an asset to Predictive Maintenance and Quality in a .json file.
2. The file is placed in the `/var/PMQ/MQSIFileInput/Insights Foundation for Energydatain` folder.
3. Predictive Maintenance and Quality message flows read the metadata, and persists the metadata as measurement types and profile variables.
4. Predictive Maintenance and Quality message flows read the master data, and persist the master data as resources and its associated static parameters.
5. Insights Foundation for Energy sends logical asset and physical asset information. A logical asset is a point where reading is captured, and a physical asset is an actual asset that is tied to a logical asset. Static properties are associated to physical assets.
6. Predictive Maintenance and Quality persists both logical and physical assets as resources. Physical assets have a parent resource as a logical asset.
7. Predictive Maintenance and Quality performs all analysis on physical assets.
8. Any errors that occur while processing metadata or master data are placed in the `/var/PMQ/MQSIFileInput/error` folder.
9. Insights Foundation for Energy does not send language and tenant information to Predictive Maintenance and Quality.

10. All Insights Foundation for Energy data that is loaded in Predictive Maintenance and Quality is in the default language and default tenant.

Load measurement data

Predictive Maintenance and Quality uses the following process to load measurement data.

1. Insights Foundation for Energy sends measurement data of an asset to Predictive Maintenance and Quality in a csv file.
2. The file is placed in the `/var/PMQ/MQSIFileInput/ifedatain` folder.
3. Insights Foundation for Energy reports measurement data for a logical asset.
4. Predictive Maintenance and Quality queries the active physical asset that is linked to the logical asset, and processes the measurement data for the physical asset.
5. A Predictive Maintenance and Quality message flow transforms Insights Foundation for Energy measurement data to Predictive Maintenance and Quality standard events. The event type is FEATURE. All Feature-Based Analytics orchestration rules are executed.
6. Any error that occurs when measurement data is processed is placed in the `/var/PMQ/MQSIFileInput/error` folder. Any error that occurs when events in Predictive Maintenance and Quality are processed are written to the `foundation.log` file.
7. Insights Foundation for Energy does not send tenant information to Predictive Maintenance and Quality. All measurement data loads that are done in Predictive Maintenance and Quality for Insights Foundation for Energy is for the default tenant.

REST Service to query predicted results

Predictive Maintenance and Quality uses the following process to query predicted results.

1. Predictive Maintenance and Quality hosts a REST service to give predicted results to Insights Foundation for Energy.
2. Insights Foundation for Energy sends REST input in .json format.
3. Insights Foundation for Energy sends profile codes and the asset class information to Predictive Maintenance and Quality, and Predictive Maintenance and Quality obtains the results for all assets that belong to that asset class. Asset classes are modeled as resource sub types in Predictive Maintenance and Quality.
4. The REST service is enabled for security. Insights Foundation for Energy sends an HTTP authorization header to send the user credentials.
5. Insights Foundation for Energy sends logical asset IDs to Predictive Maintenance and Quality in the result data, because Insights Foundation for Energy does all processing on logical assets.
6. Any errors that occur while querying the data are logged to the `/var/PMQ/MQSIFileInput/log/foundation.log` file.

Analytical content for the energy and utilities industry

IBM Insights Foundation for Energy computes relevant features for the energy and utilities industry.

The features are incorporated within Feature-Based Analytics in IBM Predictive Maintenance and Quality. The various features at an event, such as KPI and lifetime grain, allow aging predictions. The features can also be used in the Feature-Based Analytics architecture to predict health scores.

The following features are supported:

- Specific feature computations to support repair-rehab-replace. The computations are incorporated through Analytics Solutions Foundation and IBM SPSS batch jobs.
- Features for substation transformers, distribution transformers, poles, and cables are covered.
- Current and projected aging for distribution transformers are provided.

Substation transformers

Predictive Maintenance and Quality incorporates individual and aggregate features, computed by IBM Integration Bus, from the following different analysis on substation transformers. The features are used as inputs in the Feature-Based Analytics model for predicting health scores.

Overload Analysis

Analyzes Load Power and derived current to determine whether they have reached Normal or Emergency overload limits and the total overload percentage based on the season, corresponding ratings, and operating voltage.

Oil Temperature Analysis

Determines whether Top Oil Temperature and Load Tap Changer Oil Temperature have reached critical levels.

Cooling Stage Analysis

Determines whether the Load Power has exceeded the Power Limit, with Cooling based on the current Cooling Stage and the corresponding stage ratings.

Dissolved Gas Analysis

Analyzes dissolved gas content of Acetylene (C₂H₂), Ethylene (C₂H₄), and Methane (CH₄) to determine the thermal region, indicating whether the condition is critical.

Distribution transformers

Predictive Maintenance and Quality incorporates features, computed in batch mode by IBM SPSS, from the following different analysis on distribution transformers. The features are used as inputs in the Feature-Based Analytics model for predicting health scores.

Overload Analysis

Analyzes Load Energy to determine cumulative overload hours and percentage, counters of per reading or multiple hourly overload conditions that are based on the season, rated KVA, and distribution transformer sub type.

Current Aging

Gives a percentage loss life (current aging measure), based on Overload Analysis and other scientifically derived features.

Projected Aging

Gives projected aging of a distribution transformer for future years (the

start year, number of future years, and interval size is configurable), based on Overload Analysis, current aging, and user-configurable degradation factors.

Poles

Predictive Maintenance and Quality incorporates features, computed by IBM Integration Bus, based on the following parameters and status. The features are used as inputs in the Feature-Based Analytics model for predicting health scores.

Static/Lifetime parameters

The parameters include Pole Height, Reclaimed Status, Pole Age, Preservative Kind, and Species Type.

Last inspection status

The status includes inspection data such as inter-failure time (the time between inspections or failure), and various other pole status from the last inspection, such as damage, pole strength, and decay.

Cables

Predictive Maintenance and Quality incorporates features, computed in batch mode by SPSS, from the following different analysis on cables. The features are used as inputs in the Feature-Based Analytics model for predicting health scores.

Overload Analysis

Analyzes weighted Load Energy from connected distribution transformers to calculate Instant Current, based on Operating Voltage. Adds up the normal, emergency, and critical counters, in addition to hours.

Data model changes

There are data model changes in IBM Predictive Maintenance and Quality for IBM Insights Foundation for Energy.

To integrate Predictive Maintenance and Quality with IBM Insights Foundation for Energy, the following tables were added to the Predictive Maintenance and Quality data model.

- The DGA_CATEGORY table is used to process dissolved gas inputs of substation transformers. DGA_CATEGORY contains information about various thermal region categories.
- The MASTER_RESOURCE_CONNECTION table is used to hold grid connectivity information between cables and other assets, such as distribution transformers and substation transformers.

Predictive Maintenance and Quality master data loading flows are used to load data into the MASTER_RESOURCE_CONNECTION table.

Orchestration rules

There are orchestration rules in IBM Predictive Maintenance and Quality for IBM Insights Foundation for Energy.

To integrate Predictive Maintenance and Quality with IBM Insights Foundation for Energy, the following tables were added to the Predictive Maintenance and Quality data model.

Orchestration rules for substation transformers

For substation transformers, orchestration rules are configured for measurements such as Oil Temperature Top, Oil Load Tap Changer, Load Power, Cooling Stage, and Dissolved gas inputs. Dissolved gas inputs include CH₄, C₂H₄, and C₂H₂.

Features are computed during real time when raw data for the measurements is reported. To compute the features, lifetime parameters that are associated to substation transformers are also used.

Raw data and computed features are fed to the Feature-Based Analytics model to obtain the health score.

Orchestration rules for distribution transformers

For distribution transformers, orchestration rules are configured for the Load Energy measurement type and certain lifetime parameters that are associated with a distribution transformer.

Load Energy raw data is stored in the EVENT_PROFILE table, and is used by IBM SPSS batch jobs to derive computed features.

Two SPSS models are developed for distribution transformers:

- Current Aging
- Projected Aging

Predictive Maintenance and Quality uses batch orchestration to trigger the SPSS batch jobs.

Current aging results are fed as features to the Feature-Based Analytics model to obtain the health score.

Orchestration rules for poles

For poles, orchestration rules are configured for lifetime parameters and inspection data of the pole.

Inspection data and certain computed features, such as AgeInMonths and InterFailureTime, are fed to the SPSS Feature-Based Analytics model to obtain the health score.

There is no measurement data for poles, and the pole health is computed on timely basis.

In Predictive Maintenance and Quality, pole health is computed every 30 days through a message flow that uses batch orchestration.

The orchestration rules are configured in the PMQ_orchestration_definition_batch.xml file, as shown in the following figure. Users can change scheduled time and duration in days as required.

```

<!-- Orchestration for Pole FBA -->
<orchestration>
  <Identifier>POLEFBA</Identifier>
  <scheduler>
    <scheduled_time>00:00:00</scheduled_time>
    <queue_name>PMQ.POLETIMER.IN</queue_name>
    <duration_in_days>30</duration_in_days>
  </scheduler>
</orchestration>

```

Figure 27. The *PMQ_orchestration_definition_batch.xml* file

Orchestration rules for cables

For cables, orchestration rules are configured for lifetime parameters and certain derived features.

There is no measurement data for cables. The measurement data for the distribution transformer that is connected to cables is used by the SPSS model to compute cable features.

An SPSS cable job is triggered as part of the SPSS distribution transformer current aging job, and there is no separate invocation by a message flow.

SPSS results for cables are sent in the form of events, with the event type FEATURE.

SPSS generated events are added to the `/var/PMQ/MQSIFileInput/integrationin` folder, and processed by Predictive Maintenance and Quality standard event processing flows.

SPSS results are fed as features to the SPSS Feature-Based Analytics model to obtain the health score.

Managing the Standard Operating Procedures

A Standard Operating Procedure (SOP) is a set of instructions that describes all the relevant steps and activities of a process or procedure.

When you define an SOP, you define activities that are included in the SOP. IBM Insights Foundation for Energy enables an administrator to organize personnel, information, and tasks in response to events and incidents in order to achieve a comprehensive control of the operation. A SOP comprises of these components:

Standard Operating Procedure Definition

An SOP definition is the template that is used when a SOP is instantiated in response to a particular occurrence. A SOP Definition is made up of activities that are described by Activity Definitions.

Activity Definition

A SOP Definition contains one or more Activity Definitions. An activity definition sets the individual instructions that need to be performed as part of the SOP.

SOP Instance

A single Instance of an SOP in response to a particular event or occurrence. One SOP Definition can be used for many SOP Instances. An SOP Instance can be in one of these states.

- Active
- Started
- Stopped
- Completed
- Canceled

Activity Instance

An Activity Instance is the instantiation of a single Activity Definition. A single Activity Definition can be used to create multiple Activity Instances. An Activity Instance can be in a number of states:

- Active
- Waiting
- Started
- Skipped
- Completed

References

Supplemental information which is relevant to a Standard Operating Procedure or Activity. References can also be used to define e-mail templates.

Roles There are two abilities, Owners and Readers. These can be set against administrative and user roles.

- A Reader can monitor the activities that are associated with a standard operating procedure.
- An Owner can monitor and complete the activities that are associated with the standard operating procedure.

Activity Type

The Activity Type describes the response to the activity. The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

- Manual: This type of activity must be manually carried out by the owner of the SOP.
- If-Then-Else Activity: A conditional activity that allows branching based on specific criteria. The user can choose which of the SOP definitions to instantiate when starting the activity. Either enter or select values for Then and Else.
- Alert Activity: This activity displays an e-mail template for the SOP owner to complete and send an email notification to predefined personnel.
- REST Activity: An activity that creates a REST service call. The user can specify the service URL and any required authentication information to be invoked when the activity is started.
- SOP Activity: An activity that starts another standard operating procedure.

Roles for Standard Operating Procedures

The abilities for each of the roles for SOPs are as follows:

SOP Administrator roles

- View and delete an SOP definition
- Launch, view, and edit an SOP instance
- Start and complete activities in an SOP instance

SOP author roles

- Create, edit, view, and delete an SOP definition
- Create an SOP draft
- View, edit, and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft

Reference Librarian Role

Create shared references

Owner Roles (SOP definition)

- Create an SOP draft
- View, edit, and delete an SOP definition
- Edit and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft
- Launch, view, and edit an SOP instance

Reader roles (SOP Definition)

- View an SOP definition
- View an SOP instance from My Activities
- View an SOP activity, provided the user has Reader role in the Activity definition

Owners roles (SOP activity)

- View an SOP instance from My Activities
- Start and complete activities in an SOP instance for their own activities from My Activities

Reader roles (SOP activity)

View SOP instance from My Activities

Approval life cycle for an Standard Operating Procedure

An SOP definition can assume different status during its life cycle.

- Draft: When the SOP is first created, a draft version is saved initially. From an approved version of an SOP, it is also possible to create another draft version, when it is necessary to change the SOP definition using the approved version as a base. A draft can be edited, submitted for approval, or discarded.
- Pending approval: This is a draft SOP definition submitted for approval, ready to be approved or disapproved. The name of the version is defined in this status and it will name the SOP definition version if approved. If this version is not approved, the SOP definition goes back to the draft version status.
- Approved: When an SOP definition is approved it is ready to be launched.

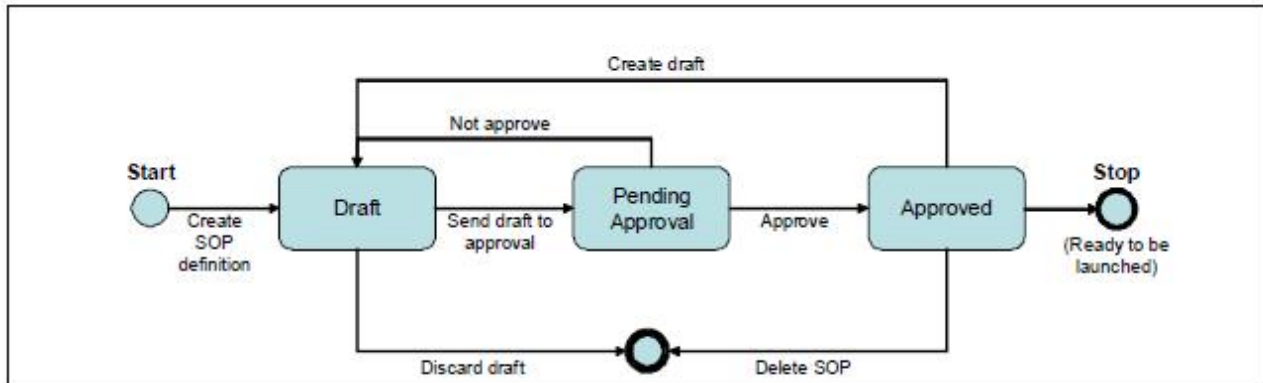


Figure 28. SOP Life Cycle

Configuring Standard Operating Procedures for different activities

One of the important tasks for when defining a Standard Operating Procedures (SOP) is to define the set of activities that composes the procedure.

The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

Configuring activities to start in sequence

When you specify that activities are done in sequence, you will not be able to start an activity until the predecessor activity is completed.

Whether the activities of an SOP are executed sequentially or not is specified in the General Settings section of the SOP definition.

Specify that activities must be executed sequentially where the activities must be executed in a chronological order or where activities depend on the result of previous activity.

Examples of sequential activities are:

- MA1 - Approve an operation.
- SA2 - Collect information about and incident area and the number of victims.
- SA3 - Prepare personnel for standby.
- SA4 - Redirect traffic to clear an incident area.

The Sequential activities figure shows an activity flow with activities that run in sequential order.

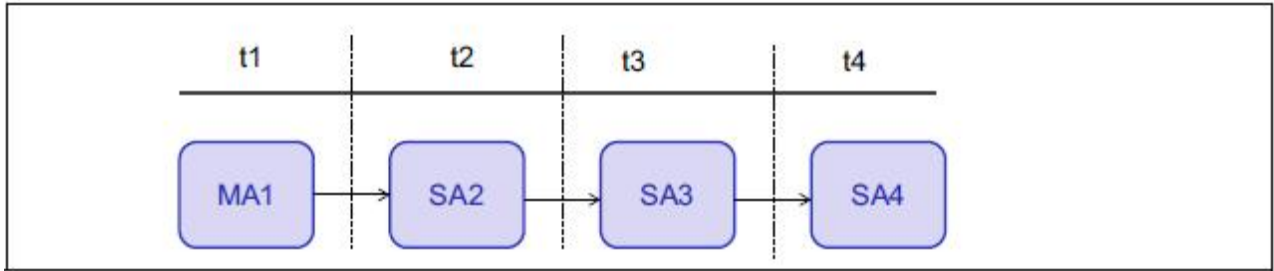


Figure 29. Sequential activities

Note:

- The user can start SA2 only if MA1 is complete, SA3 after SA2 is complete, and SA4 after SA3 is complete. The sequential property of an SOP applies to all activities, either all activities are sequential or none for the entire SOP.
- t1, t2, t3 and t4 represents the duration of the activity.

Configuring required activities

A required activity is one that is mandatory in a Standard Operating Procedure (SOP). You cannot skip over an activity that is specified as a required activity in the SOP definition.

Any of the activity types can be configured as required. Required is an attribute of an activity therefore an SOP can have activities that are required and activities that are optional (they can be skipped over). Examples of required activities are:

- MA1 - Approve operation. Mandatory.
- A2 - Collect information about incident area and number of victims. Optional.
- A3 - Prepare personnel for standby. Optional.
- A4 - Redirect traffic to clear incident area. Optional.

The mandatory activities figure shows an activity flow with activities that have mandatory and optional activities.

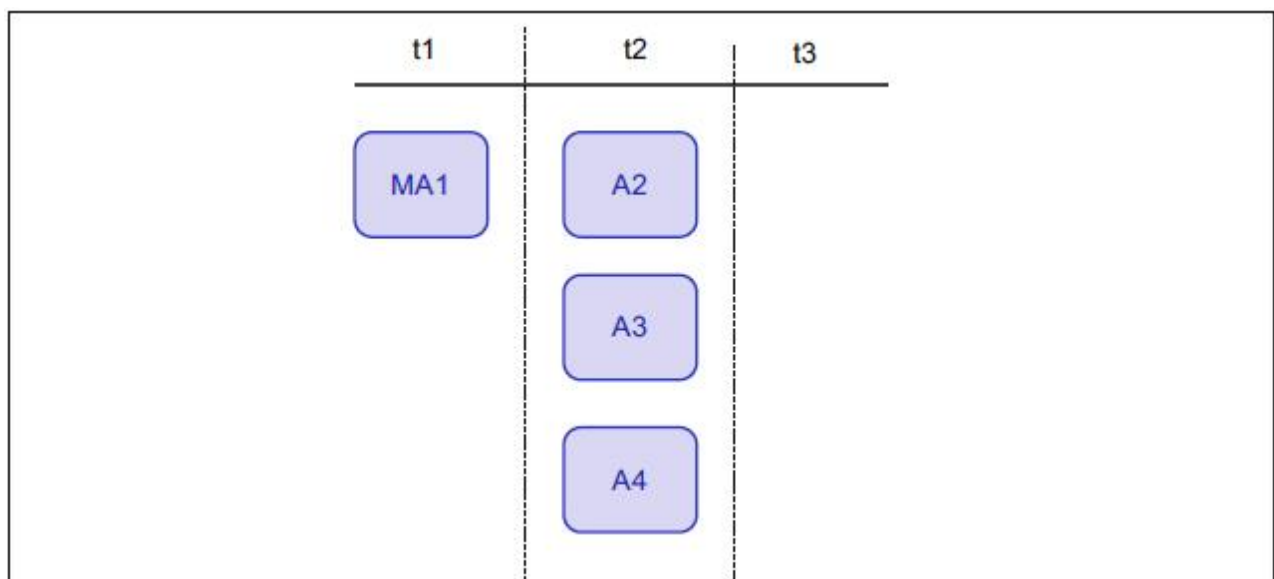


Figure 30. Mandatory activities

Note:

- MA1 is mandatory. All the other activities are optional and do not need to be performed sequentially.
- A2, A3, and A4 can be run in parallel.
- t1, t2, t3 represents the duration of the activity.

Configuring manual activities

A manual activity type is an activity that must be manually performed by the owner once the Standard Operating Procedure is launched.

Manual activities are the most basic and essential type. Examples of manual activities are:

- A1 - Collect information about the incident area and number of victims.
- A2 - Prepare personnel for standby.
- A3 - Redirect traffic to clear incident area.

The manual activities figure shows an activity flow with activities that are started manually.

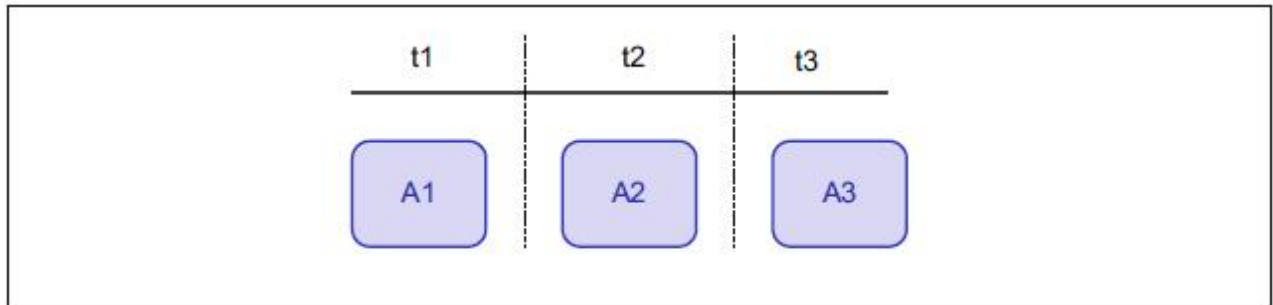


Figure 31. Manual activities

Note: A1, A2, and A3 are all manual activities. t1, t2, t3 represents the duration of each activity.

Configuring conditional activities

Conditional activities are conditional activities that allow branching based on If-Then-Else criteria.

When the activity is launched, the operator decides whether to execute one set of actions (Then) or another set (Else).

The conditional activity is useful in situations where you want to use the results from other activities to make a decision and launch a new SOP to respond to the situation. Examples of a conditional activity is:

- MA1 - Approve operation.
- SA2 - Collect information about incident area and number of victims.
- SA3 - Prepare personnel for standby.
- SA3 If (team is not available = true) Then initiate contingency plan. Conditional activity.
- SA4 - Redirect traffic to clear incident area.

The conditional activity figure shows a conditional activity flow.

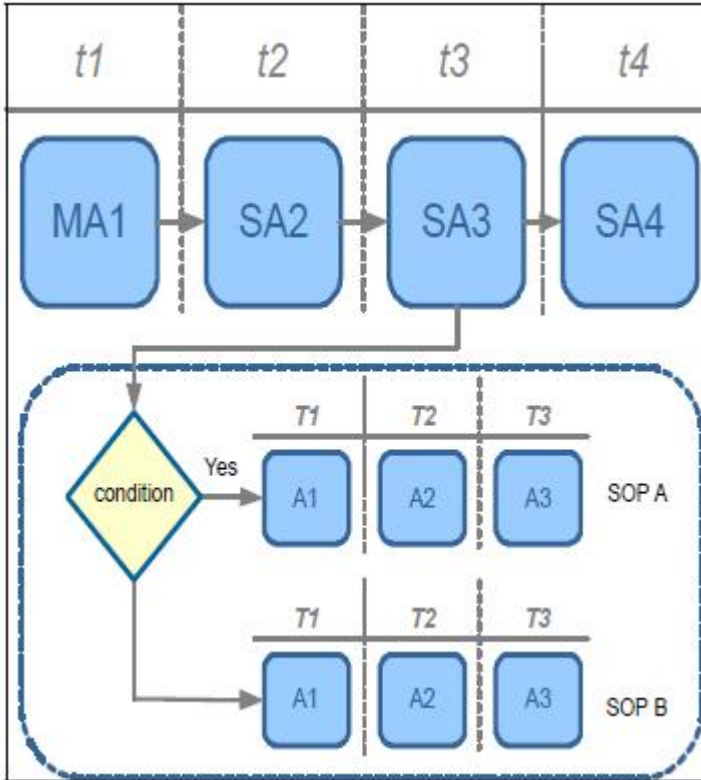


Figure 32. Conditional activity

When the SOP instance runs, the operator decides whether to launch the Then SOP or the Else SOP, if one is available, by selecting **Start Then** or **Start Else** options in my activity page.

Configuring notification activities

Notification activities enable the operator of IBM Insights Foundation for Energy to complete an email and send it. The email notification occurs as part of the required activity.

You need to set up an SMTP server to send the email notifications before configuring and launching a notification activity.

To set up an SMTP server, go to the `sysprop.json` table in the IFE database and update the following system properties in the SOP group:

- MailServerHostname: hostname of the SMTPserver
- MailServerPort: SMTP server port
- MailSender: sender of emails to be sent by the activity in the **From** field.

Email templates can be created and then reused for a notification activity. All email templates are stored as **References**.

Here are example templates There are some email examples created as references, please access them using these urls where <liberty server ip> is the ip address of the IFE server:

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_fireEvent.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_bombThreat.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_disturbanceEvent.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_evacuation.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_prep_for_power_loss.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_radiationHazard.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_transitionToNewERlevel.txt`

`https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/
emailTemplate_weatherEvent.txt`

To create a notification template:

1. Create a text file that contains the email template. The To, Subject, and Body fields are optional. Here is an example template:

```
-----  
To:  
Subject:Attention: Emergency incident detected  
Body:Emergency incident detected. Please contact administrator for  
further Information.  
-----
```

2. Publish the template in a location that accessible from IBM Insights Foundation for Energy.
3. Create a reference in the Standard Operating Procedures References and point the address to the URL where the template is published.
4. Make sure the first line of the description field contains only the keyword NOTIFICATION.
You can add further description details in subsequent lines.

Note: If you receive a 401 error when you try to load the template into the email window, you need to include your credentials in the URI, for example:`https://user_name:password@<host>:<port>/test-url/notification`.

Defining a Standard Operating Procedure

Define an SOP with all activities.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.
2. In the Basics section, define the basic information for the SOP.
 - a. Click **Create**.
 - b. In the Basics section, type a name and description for the SOP.

- c. If the activities need to be carried out in a particular order, select **Activities are done in order** in the **General Settings** field.
 - d. Click **Add References** to add supplemental information.
 - e. Click **Next** to go to the Roles section.
3. In the Roles section, define the roles for the SOP.
 - a. Select the **Roles** that will be assigned as **Owners** and **Readers** for the SOP created with this definition.
 - If you want the role to be able to monitor activities that are associated with the standard operating procedure, select **Reader**.
 - If you want a role to be able to monitor and complete activities that are associated with the standard operating procedure, select **Owner**.
 - b. Click **Next** to go to the Activities section.
4. In the Activities section, define the activities for the SOP.
 - a. Click **Add**.
 - b. If the activity is required, select **Required**. If not, the SOP can move on to the next activity during instantiation.
 - c. Select **Autostart** if this activity is to start automatically without owner operation.
 - d. You can select the roles for the Owners and Readers. If not, the roles are inherited from the SOP definition.
 - e. Set the duration of the activity. The duration is the length the activity takes once started.
 - f. Type a description of the activity.
 - g. Set the Activity type. The options are:
 - Manual Activity
 - If-Then-Else Activity
 - Alert Activity
 - REST Service
 - SOP Activity
 - h. To add more activities click **Add** and define the next activity for the SOP.
 - i. Click **Next** to go to the Summary section.
5. Review the SOP.
 - a. Review the information in the summary.
 - b. Click **Save**.

Important: The SOP is in now draft state. The SOP must be approved before it can be instantiated.

In the draft state, the SOP can be edited or deleted if not required.

Creating a reference for Standard Operating Procedures

References are supplemental information relevant to an SOP or an activity. References can also be used to define e-mail templates.

About this task

A user can create references for Standard Operating Procedures.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > References**.
2. Click **Add**.
3. In the **Name** field, type the name of the reference.
4. In the **URI** field, type or paste the web accessible address.
5. In the **Description** field, type a description of the reference.
6. If you want to restrict the use of the reference, select the **Private** check box. Otherwise the reference can be shared to other users.

Editing a Standard Operating Procedure

Edit an Standard Operating Procure.

About this task

To be able to edit an approved Standard Operating Procedure (SOP), the SOP must be returned to a draft version.

Note: You can always edit an SOP in a draft version.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.
2. From the list of SOP definitions, select the SOP to be edited.
3. From the **Version** field, select the version to be edited.
4. Click **Create Draft**. The SOP state is now in draft.
5. Click the SOP you want to edit.
 - To add an activity, click **Add** in the Activities tab.
 - To edit an activity, click the activity and then click **Edit**.
 - To delete an activity, click the activity and then click **Delete**.
 - To edit a role, select the Roles tab and then click **Edit**.
 - To edit references, select the References tab, click **Edit** or **Delete** for a specific reference or click **Add** to add a new reference.
6. When the edits are complete, click **Save**.

What to do next

Before the edits are accepted, you must submit the draft version for approval.

Submitting a draft Standard Operating Procedure for approval

You can submit a draft version of an SOP for approval or discard it.

Before you begin

The SOP must be set to Draft version, and you must have the administrator role.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.

2. Select the SOP for submission that is in **Draft** state, and click **Submit for Approval**.

Attention: An SOP that has been submitted for approval cannot be edited. The administrator can either Approve or Disapprove the SOP.

Testing a Standard Operating Procedure

Test an approved version of a Standard Operating Procedure (SOP).

About this task

An SOP must be in an approved state before it can be tested.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures**.
2. Click on the approved SOP Definition that you want to test.
3. Click **Definition Actions** and from the drop-down list select **Launch**.
Once launched, the SOP Definition list will update to indicated the number of Active SOPs based on this definition.
The **My Activities** banner indicator shows that there are one or more activities that require attention.
4. You can perform the SOP and check for completeness.
5. If the SOP needs further work, click **Create Draft** to return the SOP to a draft state.

Exporting a Standard Operating Procedure

You can export your Standard Operating Procedure definitions as an XML file. This capability is useful for migration purposes.

About this task

To export an SOP definition performs the following steps:

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.
2. Select **Standard Operating Procedures Definition** administration page.
3. From the **Definition Actions** drop-down list, select **Export All**.
4. In the pop-up window that is displayed, click **Save File > OK**.
5. Navigate to the directory where you want to save the SopDefinitions.xml file and click **Save**.

Importing a Standard Operating Procedure

You can import a previously exported Standard Operating Procedure definitions file. This capability is useful in cases where the organization has predefined SOPs that were implemented using a different tool.

About this task

To import an SOP definition performs the following steps:

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.
2. Select **Standard Operating Procedures Definition** administration page.
3. From the **Definition Actions** drop-down list, select **Import**.
4. In the Import Standard Operating Procedures window, select the XML file to upload.

The imported definitions are displayed in the list of SOP definitions. You can import an SOP definition that has the same name as an existing SOP definition. In this case, the existing SOP definition is not overwritten. Instead, two SOP definitions with the same name are displayed in the list of definitions.

5. Click **Import**.

Reverting to a particular version of a Standard Operating Procedure

You can revert back to a particular version of a Standard Operating Procedure (SOP).

About this task

The Standard Operating Procedures shows the latest version of a definition. If you need to revert to an earlier version of a definition, you can use these steps.

Procedure

1. In Insights Foundation for Energy, click **Administration > Standard Operating Procedures > SOP Definition**.
2. From the list of SOP definitions, select the SOP to be edited.
3. From the **Version** field, select the version to be reverted to.
4. Click **Create Draft**. The SOP state is now in draft.

What to do next

Before the revert is accepted, you must submit the draft version for approval.

Viewing a Standard Operating Procedure

You can view a Standard Operation Procedure (SOP) as an administrator.

There are two ways to view an SOP instance:

- From the SOP Administration page.
- From the My Activities widget.

Viewing a Standard Operating Procedure as an administrator

You can view a Standard Operation Procedure (SOP) as an administrator.

About this task

To view the SOP instance from the SOP Administration do the following steps:

Procedure

1. In Insights Foundation for Energy, click **IFE Administration > Standard Operating Procedures**.

2. Click on the approved SOP Definition that you want to view.
3. Click **Definition Actions** and from the drop-down list select **Launch**.
Once launched, the SOP Definition list will update to indicated the number of Active SOPs based on this definition.
4. Click the approved SOP Definition.
5. Click the Instances tab.
6. Click the SOP instance to view the details.

Viewing a Standard Operating Procedure as a user

You can view a Standard Operation Procedure (SOP) as a user.

Before you begin

The Standard Operating Procedure must have been launched before a user can view the instance.

About this task

Users that are not authorized to view the Standard Operation Procedures Definition administration page, can view an SOP instance from My Activities.

Procedure

1. In Insights Foundation for Energy, click **Administration**.
2. Click the **My Activities** icon.
3. Click the SOP instance you want to view.

Chapter 8. Predictive models

Use predictive models to generate the information you need to make informed operational, maintenance, repair, or component replacement decisions.

This section describes the steps that are needed to build predictive models in the predictive maintenance area by using IBM Predictive Maintenance and Quality (PMQ). It also covers some sample use cases in the manufacturing field. Later, it highlights the steps involved, starting from the business/data understanding to deploying the predictive models built for a given use case.

The following models form the basis of the predictive models in IBM Predictive Maintenance and Quality:

- The Maintenance predictive model
- The Sensor Health predictive model
- The Top Failure Reason predictive model
- The Integrated Health predictive model

Sample predictive models are provided. For more information, see “IBM SPSS artifacts” on page 218.

The training and scoring process

The steps for training and scoring the predictive models are as follows:

1. The modeling node estimates the model by studying records for which the outcome is known and creates a model nugget. This is referred to as training the model.
2. The model nugget can be added to any stream with the expected fields to score records. By scoring the records for which you already know the outcome (such as existing customers), you can evaluate how well it performs.
3. After you are satisfied that the model performs acceptably well, you can score new data (such as health score of an asset or life time of an asset) to predict how they will perform.

Optimized recommended actions

When an asset or a process is scored and identified as having a high probability of failure, recommendations can be generated.

Define recommended actions by using rules in IBM Analytical Decision Management. Use IBM Analytical Decision Management to understand the drivers that are used to define the rules, and to determine what happens based on the scores received. For example, if a score breaches a threshold, what is the resulting action? You can automate alerts for recommended actions by integrating with other systems or by defining a routing rule to send emails. Depending on the manufacturing execution systems (MES) that you use, the recommendation may be acted on automatically. You can also predict the success rate of the corrective action that is based on previous actions.

When IBM Predictive Maintenance and Quality generates recommendations, for example, to inspect an asset, you can configure the system so that the

recommendation results in a work order that is created by IBM Maximo. The work order is populated with the information needed to complete the task, for example, a device identifier and a location.

Prioritize application template

Use the prioritize application template when you have a good understanding of the predictive analytics scores and the interaction between the predictive scores. You can use the template `OptimizedAssetMaintenance.xml` to prioritize your business objective that is based on, for example, profit maximization, or downtime minimization.

The Maintenance predictive model

The Maintenance predictive model helps you optimize your Preventive Maintenance System.

In the past, a scheduler would optimize a plant's Preventive Maintenance System (PMS) by carefully altering the days that were allotted for maintenance in the OEM's default schedule. The IBM Predictive Maintenance and Quality Maintenance predictive model helps you to optimize your maintenance schedule using predictive analysis.

Often, in a new setup of PMQ sensors in the plant, even if sensor data has not gained optimal maturity for effective predictions, there may be enough data in the plant's maintenance system (Maximo/ SAP-PM etc.) to initiate a Predictive Maintenance regime. IBM PMQ's Maintenance Analytics can work on such maintenance work orders alone and does not depend on any sensor data. Therefore, the Maintenance model may help to expedite the ROI of any Predictive Analytics system before any useful sensor data is obtained.

For some resources or instances, sensor analytics alone might not provide the most accurate predictions. In this case, you can combine the results of both Maintenance Analytics and Sensor Analytics (via Integration Analytics module) to produce more optimal end results.

Data understanding

The performance indicator table `RESOURCE_KPI` contains aggregated values for each day. You can use it to prepare for model training and scoring.

The following figure shows the counts of various profiles in the dataset for a given resource and their percentages of the dataset.

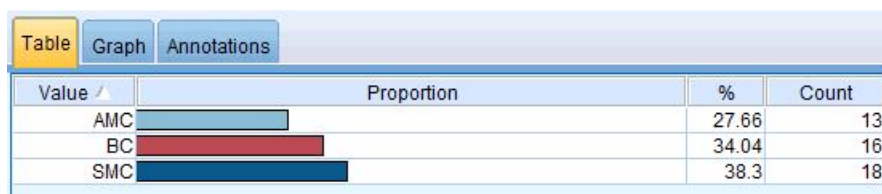


Figure 33. Percentage and counts of each profile

Additionally, the `MASTER_PROFILE_VARIABLE` and the `MASTER_MEASUREMENT_TYPE` tables help to define the appropriate codes, names, and other generic or static data.

The following figure shows a data audit node.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
KPI_DATE		Continuous	2010-01-01	2014-10-28	--	--	--	--	3287
ACTUAL_VALUE		Continuous	-82.650	70423.000	197.788	1631.452	34.824	--	2329
MEASURE_COUNT		Continuous	1	109	8.436	21.000	3.335	--	3287
PROFILE_VARIABL...		Continuous	1002	1106	1042.998	18.013	1.104	--	3287
RESOURCE_ID		Continuous	1146	1766	1174.556	122.805	4.486	--	3287
EVENT_CODE_ID		Continuous	1	1822	19.439	178.989	9.831	--	3287
LOCATION_ID		Continuous	4	1301	75.814	296.669	3.890	--	3287
PROCESS_ID		Continuous	7	73	10.654	15.097	3.890	--	3287
PRODUCTION_BA...		Continuous	11	434	34.421	96.755	3.890	--	3287
TENANT_ID		Continuous	1	1	1	0	--	--	3287

Figure 34. Data audit node

The data audit node provides summary statistics, histograms, and distribution graphs that can help you to better understand the data. The report also displays the storage icon (data type) before the field name.

Pre-modeling the data

All pre-modeling required by Maintenance Analytics is done during the MAINTENANCE.str modeling stream.

For information about both the modeling and pre-modeling data preparation, see “Modeling the data.”

Modeling the data

Modeling for the Maintenance model occurs during the MAINTENANCE.str stream.

See the following table for information about MAINTENANCE.str.

Table 20. The MAINTENANCE.str stream

Stream name	Purpose	Input	Target	Output
MAINTENANCE.str	Predicts forecasted days to maintenance interval of equipment based on Maximo work orders, and then converts these predictions into continuous health scores.	The Maximo (or other Plant Maintenance Systems') work orders converted into profiles for the actual, planned, and scheduled maintenance dates for Breakdown and Planned Maintenance.	<ol style="list-style-type: none"> 1. Custom Target as obtained using pre-data preparation within the stream itself. 2. IsFail 	<ol style="list-style-type: none"> 1. Forecasted days until the next maintenance for each resource and each historic and current day 2. Health score of the equipment for each day

There are some limitations that affect the Maintenance model:

- Limitations exist in the Breakdown + Planned maintenance work orders that are extracted from Maximo. As a result, these work orders are not optimal for forecasting directly. Breakdown + Planned maintenance work orders represent intermittent events, for which the default SPSS Time Series Modeling node cannot be used directly.
- Both types of the maintenance series contain censored data at either limit. For example, for the Breakdown series, we cannot identify from the given work orders what the optimal maintenance day would be to prevent a breakdown or irreversible wear. Similarly, for the Planned Maintenance work orders, we cannot identify the day on which a breakdown or irreversible wear would occur if we choose not to perform machine maintenance on the day identified by the Breakdown work orders.
- The series that we want to predict, that is, an ideal maintenance period, either does not exist or is divided into two series of planned and unplanned maintenance. Direct application of time series models, even with transfer function or multi-variate ARIMA models, might not help to solve the problem.

To overcome these limitations, IBM PMQ uses a custom application of Croston's forecasting methods for intermittent demand (patent pending). Using this method, the two work orders' dates series are converted to the difference of days and then combined into a single series (using censoring adjustments). This single series can be subsequently forecasted using available time series nodes in SPSS. In the current application, a simple method of global user-defined multiplicative factors is used. However, other more sophisticated, optimal, and customized methods can also be used.

The resulting value of the number of days until the next forecast can then be used to predict the machine's failure. Health scores can then be obtained using the raw propensity scores or the adj raw propensity/ or the confidence of the obtained predictions. These Health scores can be used directly or with standardization at each resource level. The present implementation uses standardization to get a uniform scale/ level of health scores for each resource.

Post modeling data manipulation

Post modeling for the Maintenance model occurs during the MAINTENANCE_DAILY.str and MAINTENANCE_EVENTS.str streams.

See the following table for more information.

Table 21. The MAINTENANCE_DAILY.str and MAINTENANCE_EVENTS.str streams

Stream name	Purpose	Input	Output
MAINTENANCE_DAILY.str	This is a post modeling data preparation stream for the purpose of preparing data for BI plots. This stream converts the predictions from the MAINTENANCE_TRENDS table into a format required by the Maintenance Overview Dashboard. The results are entered into the Maintenance Daily table in the DB.	The input data source contains all the records present in the Maintenance Trends table in the DB across all the days.	Only the current day's data with some transformations into the Maintenance Daily Table

Table 21. The MAINTENANCE_DAILY.str and MAINTENANCE_EVENTS.str streams (continued)

Stream name	Purpose	Input	Output
MAINTENANCE_EVENTS.str	This is a post modeling data preparation stream for the purpose of preparing data for BI plots. This stream converts the data from the MAINTENANCE DAILY table into a format required by the by IIB flows. The results are used to populate the IBM PMQ Events in the Event Observation table in the DB.	The input data source contains all the records present in the Maintenance Daily table in the DB.	A csv file (uploaded on the IIB integration in folder on the Analytics server) with the Maintenance Daily data in a format that can be used by IIB flows to populate the Events Table.

To improve the performance at the BI end and ensure a fast refresh and optimal user experience, all static computations and data manipulations (calculations and data manipulations not affected by user selection of prompts/ filters on the dashboards) were transferred to the SPSS batch jobs. These batch jobs can be run at an off-peak hour.

The later part of Maintenance.str and Maintenance_daily.str run the batch jobs and prepare the Maintenance Trends and Maintenance Daily tables.

The data from maintenance daily can be transferred back as events in a IBM PMQ acceptable event format. External applications can then access the events. Dashboards can also consume structure events efficiently, as the Overview dashboard does currently. The Maintenance_Events.str stream helps achieve this objective.

Evaluation of the model

This example illustrates how the Maintenance predictive model can be used effectively.

The following figure shows a time series plot with both predicted values and actual values. In this case, the predictions were accurate.

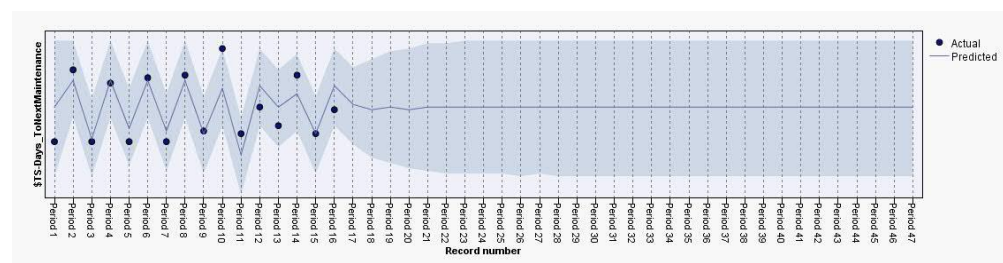


Figure 35. Time series plot

The Analysis node in the output tab helps with the evaluation of a particular model output. In this example, the predicted IsFAIL is compared on with the existing or actual values and arrived at a best-fitting training model. See the following table.

Table 22. Comparing \$L-IsFAIL with IsFAIL

Category	Value
Minimum Error	0.0
Maximum Error	1.0
Mean Error	0.032
Mean Absolute Error	0.032
Standard Deviation	0.177
Linear Correlation	
Occurrences	495

Deployment of the model

The Maintenance predictive model uses parameters from SPSS.

The model is developed using parameters which should also be used during deployment. Some parameters are configured in the downstream applications. If the parameter values are passed when the stream is executed, these values are used. Otherwise, default values are used.

The following figure shows parameters used for deployment.

Name	Long name	Storage	Value
RESOURCE_ID		Integer	1147
PROFILE_PLAN_AMC	PROFILE_VARIABLE_CD_PlannedMaintenance_ActualStart	String	AMC
PROFILE_PLAN_SMC	PROFILE_VARIABLE_CD_PlannedMaintenance_ScheduledStart	String	SMC
PROFILE_BREAKDOWN_BC	PROFILE_VARIABLE_CD_BreakdownMaintenance_Reported	String	BC
R_CENSURING	RightCensuring(Value>1)_PlannedMaintenanceLifeEnhancement	Real	1.2
L_CENSURING	LeftCensuring(Value<1)_BreakdownMaintenanceLifeReduction	Real	0.9
MAX_FUTURE_DAYS	Maximum_Future_Days_For_Which_Prediction_Is_Required	Integer	31

Figure 36. Parameters used for deployment

You can find all of these parameters using SPSS. However, only the RESOURCE_ID is exposed from the IIB end out of the box. This is because the stream has multiple execution branches that use scripts to sequence the parameters. You can see the scripts being referenced in the Execution tab.

Recommendations from ADM

The Maintenance predictive model provides scores and data that allow you to adjust maintenance dates optimally.

The deployed model, once invoked, helps to produce probability and propensity scores. However, probability and propensity scores may not be very useful to an end business user. Therefore, the results are consumed by IBM SPSS Decision Management, which then provides a more useful, text-based result.

The following figure shows the probability and propensity scores.

<input type="checkbox"/> Prepone_Maintenance_Dev_LT_-100	2005
DEVIATION_PERCENT < -100	
<input type="checkbox"/> Maintenance_as_planned_bet_0_10	3001
DEVIATION_PERCENT BETWEEN 0.0 and 10.0	
<input type="checkbox"/> Maintenance_as_planned_bet_-10_0	3002
DEVIATION_PERCENT BETWEEN -10.0 and 0.0	
<input type="checkbox"/> No Forecast Available	4001
FORECASTED_DAYS IS NULL	

Figure 37. Probability and propensity scores

Based on the scores and the data received from the modeler stream, we can determine whether specific maintenance tasks should be rescheduled.

The Sensor Health predictive model

The Sensor Health predictive model analyzes an asset's sensor readings to help determine the likelihood that the asset will fail. If the likelihood of failure is high, you can schedule an urgent inspection of the machine.

The Sensor Health model continuously monitors the health of a machine or an asset and predicts potential machine faults in real time. The model uses historical sensor data profile values stored in the KPI tables and keeps a running status to determine the current health of an asset. The Sensor Health model can also be used to predict the future health of an asset.

Tip: If there are too many failures (for example, more than 30% of the days, or multiple times in a day), then instead of using the KPI tables for training, a user could use raw events from the event table for training with appropriate filtering or treatment of noise, if any.

Data understanding

The Sensor Health predictive model uses the RESOURCE_KPI and MASTER_PROFILE_VARIABLE tables.

The performance indicator table RESOURCE_KPI is used to hold aggregated values for each day. The table can be further used to prepare for model training and scoring. The MASTER_PROFILE_VARIABLE is used to help identify the specific profiles and select only the profiles that require further analysis.

The following diagram shows an example of the source data stream for the Sensor Health predictive model.

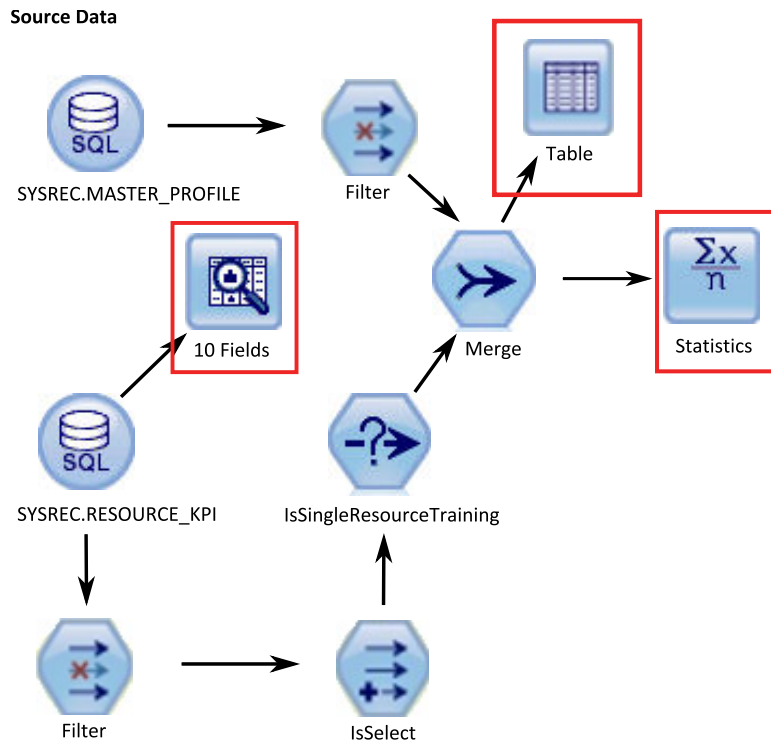


Figure 38. Example source data stream

In the diagram, the highlighted red boxes indicate possible ways that we can interpret the data. For example, the Statistics node addresses the summary statistics for individual fields and the correlations between fields. The Data Audit node provides a comprehensive first look at the data and is presented in an easy-to-read matrix. This matrix can be sorted and used to generate full-size graphs and a variety of data preparation nodes.

Data preparation

Data preparation for the Sensor Health predictive model occurs during execution of the SENSOR_HEALTH_DATA_PREP stream.

The SENSOR_HEALTH_DATA_PREP.str data preparation stream extracts the data from Predictive Maintenance and Quality tables and prepares the data to be used in the modeling. The eligible data is exported to a csv file for the modeling. The input data source contains the measurement type actual reading information of machines. The output is a list of machines for which there is enough data and that are eligible for training to identify the patterns.

To prepare for analysis of the health score based on the measurement types, only the machine's measurement type attributes are considered. Each measurement type has a value. The number of times that the value exceeds the upper and lower limits is accounted for. To train the model to identify failure patterns, enough failure data must be available. Machines that do not have sufficient failure data are not eligible for further modeling. Machine names are logged in the file Training_Eligibility_SensorAnalytics_Report.csv. In this file, resources are indicated either with 1 (eligible) or 0 (not eligible).

The following diagrams show an example of a data preparation stream for the Sensor Health predictive model.

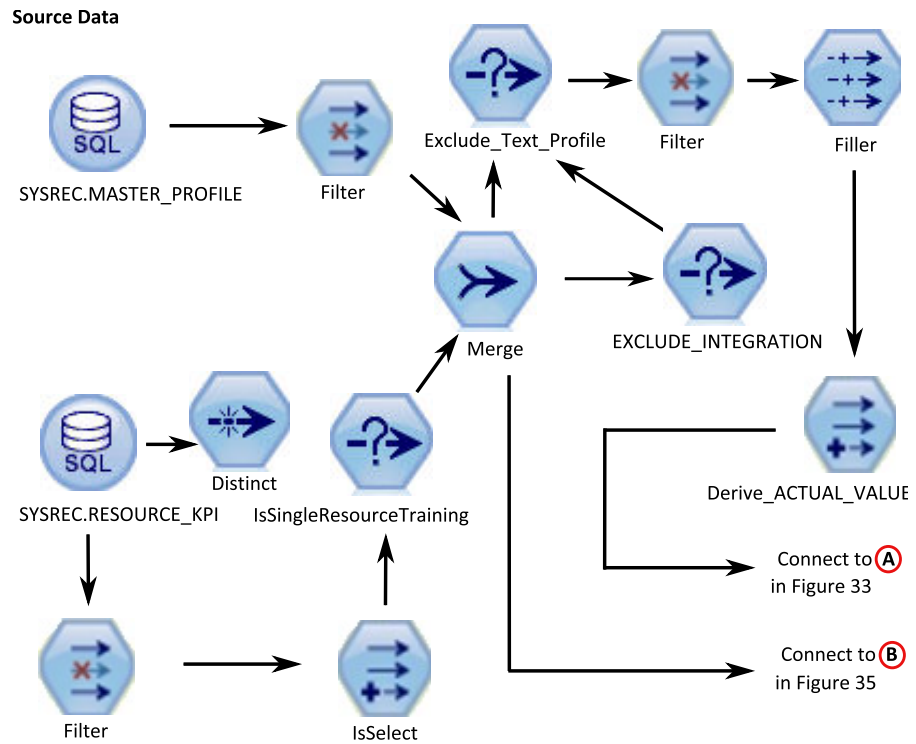


Figure 39. Example data preparation stream for the Sensor Health predictive model - Part 1

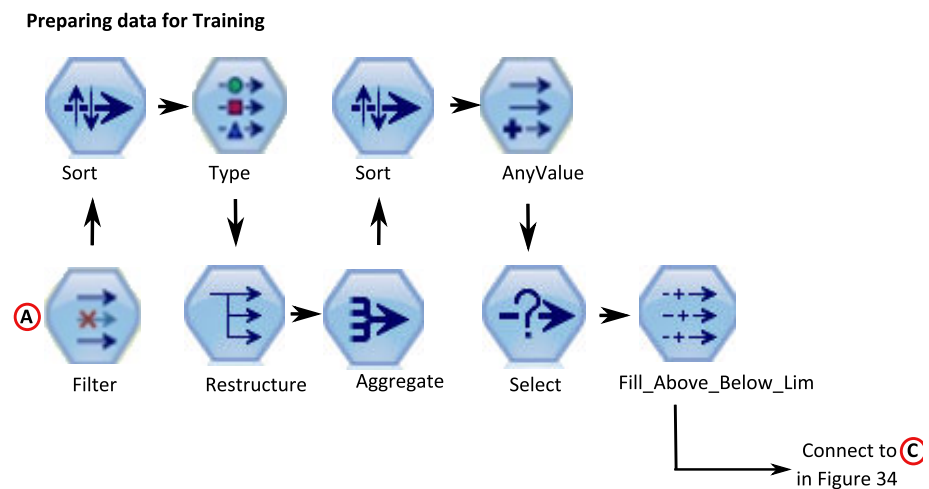


Figure 40. Example data preparation stream for the Sensor Health predictive model - Part 2

Preparing data for Training (continued)

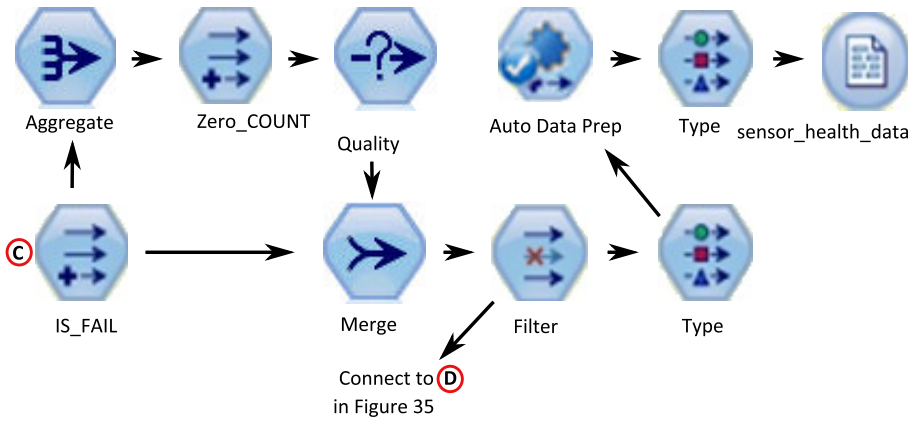


Figure 41. Example data preparation stream for the Sensor Health predictive model - Part 3

Preparing Log File

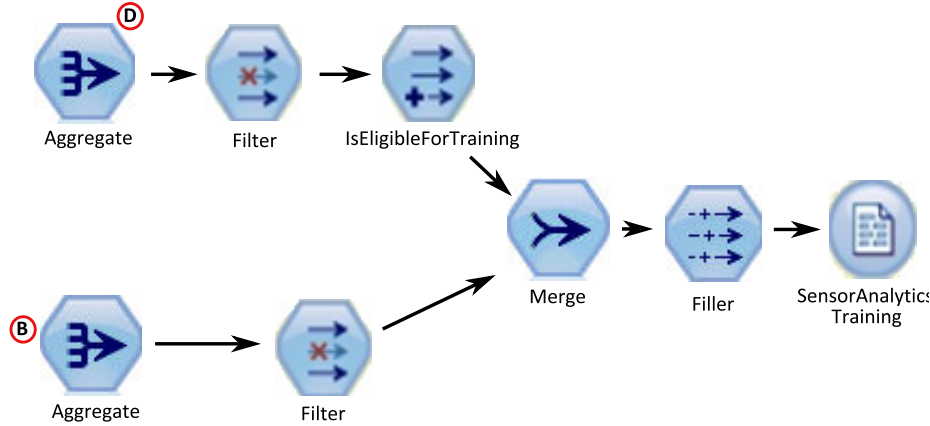


Figure 42. Example data preparation stream for the Sensor Health predictive model - Part 4

Data modeling

The Sensor Health predictive model uses the SENSOR_HEALTH_COMBINED.str stream.

See the following table.

Table 23. The SENSOR_HEALTH_COMBINED.str stream

Stream name	Purpose	Input	Target	Output
SENSOR_HEALTH_COMBINED.str	Predicts failure of equipment based on the measurement types received thru the sensor details, train the models and also refresh them for the scoring service	The machine levels measurement type data received thru the sensor reading systems	IS_FAIL	Health score of the equipment

The following figures show an example of a modeling stream for the Sensor Health predictive model.

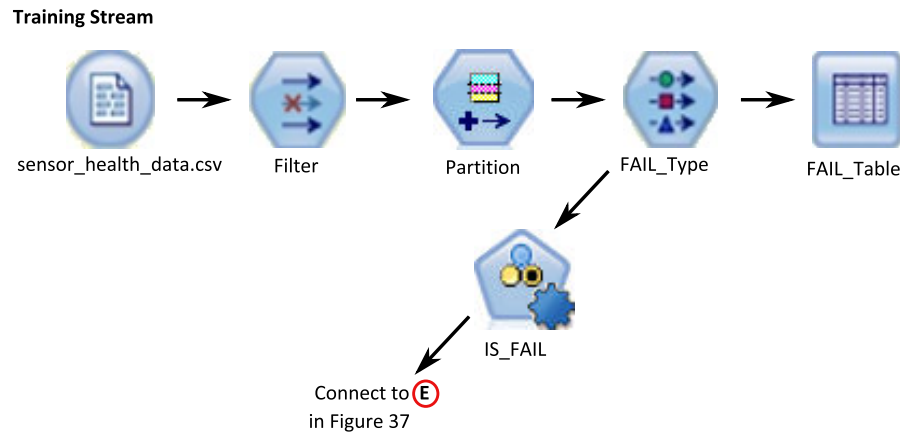


Figure 43. Example modeling stream for the Sensor Health predictive model - Part 1

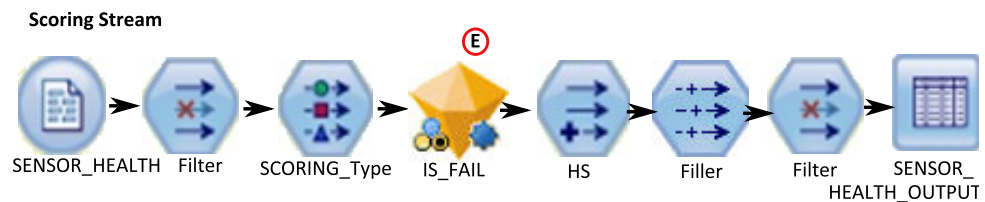


Figure 44. Example modeling stream for the Sensor Health predictive model - Part 2

Depending upon the input data, you might need to consider a different approach for the health score modeling. In addition, the concept of Splits at a resource id level (in Type Node) is introduced because, for each resource, the trained model would be unique.

The value of the health score of an asset is between 0 and 1. The higher the value of the health scores, the better the health of the asset. If the input data model and structure is modified, the health score model must be retrained on the new data.

The health score model is based on the IBM SPSS Modeler auto classification model's confidence. Alternatively, raw and adjusted raw propensity scores can be used to generate such scores. In the model node, there are options to modify the costs, revenue, and weights. This setting depends on the requirements and the available data. Similarly, the data in this case is not balanced. Depending on the data and requirements, balancing might give better results.

Note: Not all of the models support propensity score outputs, especially when splits are enabled.

Evaluation of the model

The model should be verified against the business success criteria established at the beginning of the project.

At this point, most of the data mining activities are complete. However, there is a need to verify the model across the business success criteria that were established at the beginning of the project. We asked the following questions:

- Did the Health scores that were generated from sensor readings provide any helpful insights?

- What new insights or surprises were discovered?
- Were there any issues that were caused by inadequate data preparation or misinterpretation of the data? If there was an issue, we returned to the appropriate stage and rectified the problem.

Deployment

The Sensor Health predictive model uses a combined stream that performs several functions.

The model is developed using parameters which should also be used during deployment. Some parameters are configured in the downstream applications. If the parameter values are passed when the stream is executed, these values are used. Otherwise, default values are used. See the following figure.

Name	Long name	Storage	Value
IS_1_RES_TRAIN	Resource Training required	Integer	0
RESOURCE_ID	Resource identifier	Integer	595

Figure 45. Parameters used for Deployment

If there is a provision to train one resource at a time, the resource id is passed along with the flag value.

This combined stream performs the following functions:

- helps to train the models
- refreshes data for the scoring service
- uses auto-modeling to identify the best suitable model
- produces health score output that measures the probability of machine failure

The stream has multiple execution branches that use scripts to sequence the parameters. Note that the scripts being referenced appear in the Execution tab. See the following figure.

Options Messages Parameters **Deployment** Execution Globals Search Comments Annotations

Deployment type: Model Refresh

Model Refresh: enables creation of streams and scenarios supporting scoring, building and refresh features

Deployment Settings

Scoring node: SENSOR_HEAL... Select any terminal node
 Scoring Parameters... (0/0 parameters selected)

Modeling node: IS_FAIL Select a modeling node
 Model Build Parameters... (0/0 parameters selected)

Model nugget: IS_FAIL Select a model nugget in the scoring branch

Figure 46. Refreshing the data for the scoring service

The stream is auto-generated when a training instance occurs and, for the real time scoring, in the SENSOR_HEALTH_SCORE service which is invoked by the IIB flows.

Recommendations

The Sensor Health predictive model provides recommendations for each asset.

Sensor analytics recommendations are produced using the real-time mode of invocation. In the invocation mode, the stream is developed using the ADM and a SENSOR_RECOMMENDATION service is configured for scoring services. The service is invoked to receive a recommendation for each asset. See the following figure.



<input type="checkbox"/> Urgent Inspection HS >= 0.7	HS101
<input type="checkbox"/> Need Inspection HS BETWEEN 0.4 and 0.7	HS102
<input type="checkbox"/> Remainder	HS103

Figure 47. Recommendation settings

Depending on the Health score calculated from the Modeler, a recommendation of an Urgent Inspection (HS101) may be produced. For each HS101 code, a trigger is sent to Maximo to create the work order.

The Top Failure Reason predictive model

The Top Failure Reason predictive model helps you identify the top failure predictors for a given asset in order of their importance. You can then further analyze the identified reasons or parameters to assist in a guided trail from cause or root cause analysis to its respective pattern discovery.

This model is used to analyze and discover the top percentile and number of parameters which are influential in predicting a machine's failure (or optimal health) and their relative importance.

Understanding the data

The Top Failure Reason predictive model uses the event and master tables from the IBM PMQ database to get the sensor data available for each resource at a given point in time. It also gathers the defective and failure information.

The performance indicator table RESOURCE_KPI contains aggregated values for each day. You can use it to prepare for model training and scoring. The MASTER_PROFILE_VARIABLE and the MASTER_MEASUREMENT tables are used to help identify the specific profiles which are considered as parameters and will be considered for further analysis.

Preparing the data

Preparation for the Top Failure Reason predictive model includes merging data, selecting a sample subset, deriving new attributes, and removing the unwanted fields.

Depending on the data and identified goals, in this phase of data preparation the following tasks are performed:

- Merging data sets and records of the master and the events data

- Selecting a sample subset of data, identifying only the specified resource and profiles
- Deriving new attributes for each of the selected profiles based on the parameters
- Removing unwanted fields that are not required for further analysis

The measurements used as parameters are based on the data understanding. They are kept as parameters so they can be modified later, based on the data set. In the IIB layer, only the resource id is available.

Modeling the data

The prepared data is now considered for the modeling exercise. The target is set as the IS_FAIL variable and uses the Logistic regression model to obtain a percentile or probability value.

See the following figure.

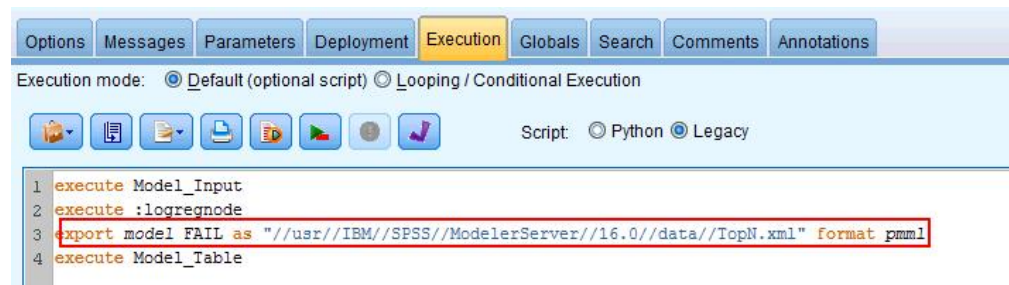


Figure 48. The Execution tab for the modeling stream

The stream has multiple execution branches that use scripts to sequence the parameters. You can see the scripts being referenced in the Execution tab. The important point here is to have exported the model FAIL in the pmml format. This is consumed in the TopN_XML stream to obtain the appropriate predictive importance of each profile.

Evaluation

The Top Failure Reason predictive model should be verified against the business success criteria established at the beginning of the project.

The Cumulative Gains chart shows the advantage of using a predictive model instead of a default, random model. The random model (depicted by a red line in the following figure) shows an equal ratio of percent gain (that is, the selection of entities of interest) to the percent of the total number of entities processed. Therefore, the red line has a 45 degree slope and the gains percentage is equal to the population percentile.

Cumulative Gains charts always start at 0 percent and end at 100 percent. In the following Cumulative Gains chart, the percentage gain rises from zero percent to one hundred percent as the percentage of failures rises from zero percent to forty percent. Continuing after the forty percent failure rate, there are no gains until one hundred percent of the assets have failed.

A good predictive model has a steeper slope than the random model. When using a predictive model, the goal is to categorize and predict more entities of interest than would be done randomly. The model shown in the following figure can

predict all the entities of interest by incorporating only 40 percent of the total population.

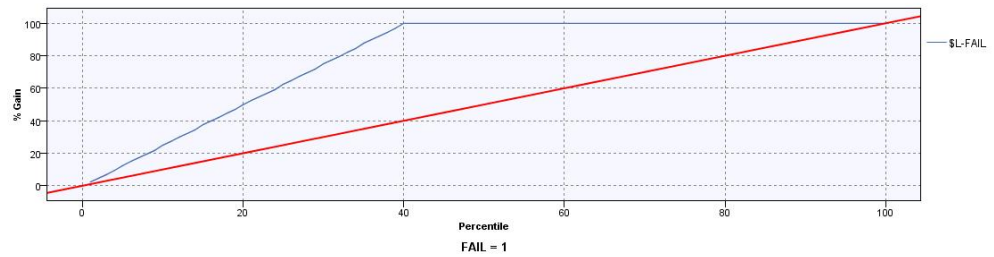


Figure 49. A Cumulative Gains chart

For example, a training and validation set contains only 2 percent of defective parts. Using a random selection model, we would have to select 100 percent of the parts to identify the intended 2 percent of the faults. However, using the model in the preceding figure, we only need to select the top 40 percent of the parts that are most likely defective. This will ensure that all 2 percent of the intended defective parts (equivalent to 100 percent of gain) are covered in our target set.

Deployment

The output of the model calculates the cumulative sum of all the predictive importance values. These values are exported to a csv file. The IIB flow loads the csv file into the profiles table that will be consumed in the Cognos charts.

Note: You can alter the values at each resource level by exposing them to IIB and making a mechanism for picking the correct parameters for each resource. Otherwise, for ad-hoc purposes, the parameters may be changed and triggered manually for each required resource. Also, the output table contents must be deleted manually for this operation, in case the data for the same resource exists from an earlier run.

The Feature-based predictive model

The Feature-based predictive model produces a resource's predicted health score and optimal maintenance period, and recommends inspection or changes to maintenance schedules.

Sample features supported in IBM Predictive Maintenance and Quality

The following list describes sample features that Predictive Maintenance and Quality supports.

- Based on the raw sensor reading of a single sensor.
For example, <Load>-<RAW> = Raw Sensor reading for Load
- Based on computations that involve a single sensor reading.
For example, <FlowMeter>-<OilFlow> = $\text{delta}(\log(\text{FlowMeter}-\text{RAW}+2))$
Subtypes under this class of features include mathematical operators, system-defined mathematical functions, logical operators, comparison operators, If Then-Else conditions, and time functions.
- Based on computations that involve more than one sensor, and/or sensor+ time stamp types.

For example, $\langle \text{FlowMeter} \rangle - \langle \text{OilFlowWeighted} \rangle = \text{Power} (\text{delta}(\log(\langle \text{FlowMeter} \rangle - \langle \text{RAW} \rangle + 2)) / \text{delta}(\text{Mean}(\langle \text{RPM} \rangle - \langle \text{RPM-FEATURE-1} \rangle + 2))$

- If Else condition based on one or more sensors and a time stamp
For example, $\langle \text{current} \rangle - \langle \text{overload} \rangle = \text{If} (\text{month}(\text{timestamp}) \text{ In } (2,3,4) \text{ and } \langle \text{Current} \rangle - \langle \text{Raw} \rangle > \langle \text{Static} \rangle - \langle \text{SummerOverload} \rangle) \text{ OR } (\text{month}(\text{timestamp}) \text{ Not In } (2,3,4) \text{ and } \langle \text{Current} \rangle - \langle \text{Raw} \rangle > \langle \text{Static} \rangle - \langle \text{WinterOverload} \rangle) \text{ THEN } 1 \text{ Else } 0$
- Based on sensor reading and nameplate parameters.
For example, $\langle \text{Static} \rangle - \langle \text{OverloadThreshold} \rangle = 0.8 * \langle \text{Static} \rangle - \langle \text{RatedKVA} \rangle$
 $\langle \text{Load} \rangle - \langle \text{OverloadFactor} \rangle = \text{Log} (0.9 * \langle \text{Load} \rangle - \langle \text{RAW} \rangle / \langle \text{Static} \rangle - \langle \text{RatedKVA} \rangle)$
- Based on nameplate parameters.
For example, $\langle \text{Load} \rangle - \langle \text{RatedVA} \rangle = \langle \text{Static} \rangle - \langle \text{RatedKVA} \rangle * 1000$
- Logical comparison and condition-based features.
For example, $\langle \text{Load} \rangle - \langle \text{WeightedOverload} \rangle = \text{If } \langle \text{Load} \rangle - \langle \text{OverloadFactor} \rangle > \langle \text{Static} \rangle - \langle \text{OverloadThreshold} \rangle \text{ then } \langle \text{Measurement-2} \rangle * \langle \text{Feature-3} \rangle * 0.75 \text{ else } 0$

Features are modeled as profile calculations. The following calculations are supported:

Comparison operators

Various comparison operators are supported.

Logical operators

Various logical operators are supported.

Date calculations

Various date calculations are supported, including Day of Month and Year.

Date math calculations

Various date mathematical operations are supported.

If Else calculations

You can define If Else conditions.

Math calculations

Simple mathematical calculations are supported, such as addition and subtraction.

Advanced math calculations

Mathematical functions are supported, such as cos, sin, log, and exp.

Measurement text does not contain count

Users can check whether an observation text matches a string.

Substation cooling stage analysis and substation overload analysis

These calculations apply to substation transformers.

Derived preservative kind, derived species type, and pole analysis

These calculations apply to poles.

Data model changes for Feature-Based Analytics

IBM SPSS requires both raw sensor readings and the computed features that are derived by using sensor readings in consistent way, from a consistent data source (table), and on same grain as event.

For Feature-Based Analytics, a new profile table, EVENT_PROFILE, is supported in Predictive Maintenance and Quality, which allows users to store profiles at the

same grain of the event. All features that are computed for the Feature-Based Analytics model are stored in the EVENT_PROFILE table.

Features that are aggregated at day and life time grain are stored respectively in the RESOURCE_KPI and RESOURCE_PROFILE tables.

SPSS also supports resource sub type level modeling. As a result, there is a new table in Predictive Maintenance and Quality, called MASTER_PROFILE_MAPPING, that holds mappings between resource sub types and profile mappings. Predictive Maintenance and Quality master data loading flows are used to load data in the MASTER_PROFILE_MAPPING table.

Orchestration rules

The data preparation for Feature-Based Analytics is done through Predictive Maintenance and Quality standard event processing flows. Events for Feature-Based Analytics must have the event type FEATURE. When feature events are processed by the orchestration engine, the following orchestration steps are executed by the specified adapter.

1. Service Adapter

This step is executed only when Dissolved Gas Analysis measurements are reported. This step is bypassed for all other measurement types. This step performs computations for gas measurement data.

2. Profile Adapter

This step computes various features on raw reading data, and stores the computed results in KPI and profile tables.

3. Scoring Adapter

This step invokes the SPSS Feature-Based Analytics model through the Representational State Transfer (REST) interface.

4. Scoring Adapter

This step invokes the SPSS Integration Analytics model through the REST interface.

5. Service Adapter

This step performs post-processing of Integration Analytics results.

6. Service Adapter

This step creates a work order in IBM Maximo if the Integration Analytics model provides Urgent Inspection as the recommendation. Work orders are created in Maximo through the Service Adapter.

Training

Feature-Based Analytics training is timer-based, and it is configured through generic batch orchestration. Generic batch orchestration provides capabilities to run the scheduler or invoke any SPSS batch job by taking inputs from a configurable XML file, instead of developing separate message flows for each use case implementation.

Feature-Based Analytics training is scheduled to trigger every 90 days. If you must schedule training earlier, modify the orchestration definition XML file. Modify the values in the <scheduler> element as shown in the the following figure.

```

<!-- Orchestration for FBA Training -->
<orchestration>
  <Identifier>FHSTrigger</Identifier>
  <scheduler>
    <scheduled_time>02:00:00</scheduled_time>
    <queue_name>PMQ.FHSTIMER.IN</queue_name>
    <duration_in_days>90</duration_in_days>
  </scheduler>
  <!-- Webservice configuration for FBA Training-->
  <webservice>
    <url>http://localhost:9080/process/services/ProcessManagement</url>
    <jobLocationURI>spsscr:///?id=5691007b1cf528e700000149562efe098639</jobLocationURI>
    <notificationEnabled>true</notificationEnabled>
  </webservice>
</orchestration>

```

Figure 50. Location in the orchestration definition file where to modify the training schedule

Input data for training

Feature-based health score and days to maintenance training requires a combination of input data.

The following list describes the input data that the feature-based model requires.

- raw sensor events
- static or nameplate parameters
- day level KPI data
- lifetime parameters
- domain-specific features, which are computations on a combination of one or more raw events, static parameters, or lifetime parameters

Sensor data and features are processed and loaded into the Predictive Maintenance and Quality data store through the Predictive Maintenance and Quality event load flow, as specified in the orchestration definition file. Feature calculations are configured in IBM Integration Bus (IIB), or implemented in SPSS batch jobs.

Minimum data requirements

The minimum data that feature-based analytics requires is three failure and three non-failure events per resource, or per resource sub type for sub type level models, with optimal data quality for other parameters.

The input data can be at event grain, day grain, or lifetime grain. For day grain data, any KPI date with even a single failure event in that day is considered a failure date. It is recommended to use noise treatments, as the training event uses data too. For event grain data, users can flag events before failure as fail events for realistic modeling.

Resources that do not satisfy the minimum data requirements are not trained, and they are logged in the Training Eligibility report as 0 against their eligibility to be trained. Any attempt to score these resources returns a health score of -1 or a predicted days to maintenance of -9999. These results are identified and dropped by IBM Integration Bus (IIB), and they are not available in databases for downstream applications.

Note: The data requirements are only an implemented bare minimum, and they do not guarantee an optimally trained model.

For resources that satisfy the bare minimum data requirements, IBM SPSS can sometimes give errors or fail some of the ensemble models. The errors or failures occur when the data quality is not good, or for other reasons. Under such conditions, the data volume and quality must be improved. As a last resort, you can remove the failing models from the expert modeler.

Resource sub type level modeling

Sub type level modeling is sometimes needed to supplement resource level modeling.

In many cases, equipment failures occur after a long duration, and they are replaced or refurbished after the failure. Because IBM Predictive Maintenance and Quality 2.0 Sensor Analytics is trained per resource, there are no trainable resources for these failures.

Users can have a custom segmentation model that will segment resources, even within a type, based on similar characteristics, and use that segment information within the model. Predictive Maintenance and Quality 2.5 supports segmentation at a resource sub type level.

During scoring, the health score and forecasted days to maintenance come from the most specific model that was trained. For example, for a resource, if the resource level model was not trained, then the resource sub type model is used for scoring. If the resource level model was trained, then that model is used for health scores and forecasted days to maintenance.

Training job

The `IBMPMQ_SENSOR_FEATURE_BASED_ANALYTICS` job calls the data preparation stream, followed by training and refreshing the health score and forecasted days to maintenance models.

The following diagram shows the `IBMPMQ_SENSOR_FEATURE_BASED_ANALYTICS` job.

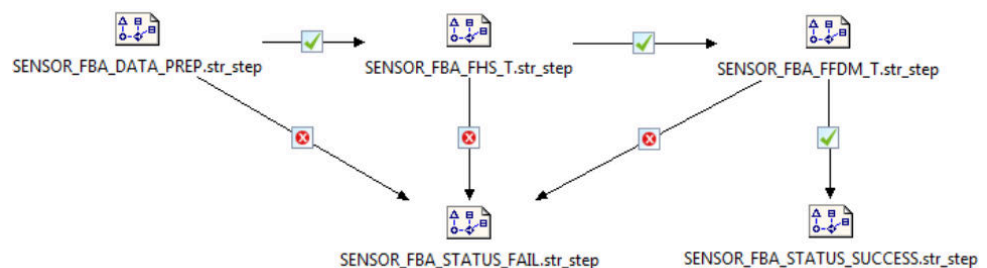


Figure 51. The `IBMPMQ_SENSOR_FEATURE_BASED_ANALYTICS` job

At the end of the job, the file `Training_FBA_in.csv` is returned to the `\root\integrationin` directory. The file contains two fields, System Timestamp and Status, which are read by the IBM Integration Bus (IIS).

If there is failure at any step in the job, the file contains `FAIL` as the status. When all steps are completed successfully, the file contains `SUCCESS` as the status.

Data preparation

Data preparation for the Feature-based predictive model occurs during execution of the `SENSOR_FBA_DATA_PREP.str` stream.

The following table describes the `SENSOR_FBA_DATA_PREP.str` stream.

Table 24. The `SENSOR_FBA_DATA_PREP.str` stream

Stream name	Purpose	Input	Output
<code>SENSOR_FBA_DATA_PREP.str</code>	A Data preparation stream extracts the data from IBM Predictive Maintenance and Quality tables, and prepares the data to be used in the modeling. The eligible data is exported to a csv file for the modeling.	The input data source contains raw sensor event data, day level and lifetime KPIs, and static or nameplate parameters with appropriate linking to the master data tables. It also references the IBM Maximo, or other plant maintenance systems, work orders that were converted into profiles for the actual, planned, and scheduled maintenance dates for breakdown and planned maintenance.	A list of machines for which there is enough data and that are eligible for training to identify the patterns. Transformed data input to modeling for the eligible resources.

This stream prepares the data for analysis of the health score and forecasted days to maintenance, based on raw sensor data, day and lifetime KPI data, and static or nameplate parameters. In addition, to train the model to identify failure patterns, enough failure data must be available.

Machines that do not have sufficient failure data are not eligible for further modeling. Machine IDs are logged in the `Training_Eligibility_SensorFBA_Report.csv` file. In this file, resources are indicated either with 1 (eligible) or 0 (not eligible).

Based on the specific domain and equipment data available, if data or features are added, deleted, or modified, then the data preparation stream must be modified to incorporate the changes. In addition, the framework processes both numeric and categorical features in the final modeling input. There is also a provision to enable or disable certain features out of the input. This provision is configured at a resource sub type level in the `MASTER_PROFILE_MAPPING` table.

Data modeling

The Feature-based predictive model uses different streams for health score modeling and for forecasted days to maintenance modeling.

Health score modeling

The following table describes the `SENSOR_FBA_FHS_T.str` stream that is used for health score modeling.

Table 25. The *SENSOR_FBA_FHS_T.str* stream

Stream name	Purpose	Input	Target	Output
SENSOR_FBA_FHS_T.str	Predicts the failure of equipment based on the measurement types that are received through the features, trains the models, and refreshes them for the scoring service	Transformed data (raw event data, day level and lifetime KPIs, and static or nameplate parameters) from the data preparation stream	IS_FAIL	Health score of the equipment

Depending on the input data, you might have to consider a different approach for the feature-based health score modeling. In addition, the concept of Splits is introduced at a tenant ID, location ID, resource sub type, and resource ID level in the Type Node, because the trained model must be unique for each combination of tenant, location, and resource.

The value of the health score of an asset is 0 - 1. The higher the value of the health scores, the better the health of the asset. If the input data model and structure is modified, the health score model must be retrained on the new data.

The health score model is based on the IBM SPSS Modeler auto classification model's confidence. Alternatively, raw and adjusted raw propensity scores can be used to generate such scores. In the model node, there are various modeling technique-specific settings. These settings can be configured based on the requirements and the available data. Similarly, the data in this case is not balanced. Depending on the data and requirements, balancing might give better results.

Note: Not all of the models support propensity score outputs, especially when splits are enabled.

Forecasted days to maintenance modeling

The following table describes the *SENSOR_FBA_FFDM_T.str* stream that is used for forecasted days to maintenance modeling.

Table 26. The *SENSOR_FBA_FFDM_T.str* stream

Stream name	Purpose	Input	Target	Output
SENSOR_FBA_FFDM_T.str	Predicts forecasted days to maintenance based on the measurement types that are received through the features, trains the models, and refreshes them for the scoring service	Transformed data (raw events, day level and lifetime KPIs, and static or nameplate parameters) from the data preparation stream	DTM (Days to maintenance based on the maintenance or failure records)	Forecasted days to the next maintenance for the equipment

Depending on the input data, you might have to consider a different approach for the feature-based forecasted days to maintenance modeling. In addition, the concept of Splits is introduced at a tenant ID, location ID, resource sub type, and resource ID level in the Type Node, because the trained model must be unique for each combination of tenant, location, and resource.

The value of the forecasted days to maintenance determines the optimal maintenance period. The lower the value, the earlier the maintenance is recommended.

The forecasted days to maintenance are based on the SPSS Modeler auto numeric model. In the model node, there are various modeling technique-specific settings. These settings can be configured based on the requirements and available data. Similarly, the data in this case is not balanced. Depending on the data and requirements, balancing might give better results.

Deployment

In its data preparation stream, the Feature-based predictive model exposes certain parameters that can be used to modify the input data that is created for modeling.

Some parameters are configured in the downstream applications. If the parameter values are passed when the stream is executed, these values are used. Otherwise, default values are used. See the following figure.

Name	Long name	Storage	Value	Type
IS_1_RES_TRAIN	Train for a Single Resource	Integer	0	(no values)
RESOURCE_ID	Which Resource to train for...	Integer	461	(no values)
PROFILE_PLAN_AMC	PROFILE_VARIABLE_CD...	String	AMC	(no values)
PROFILE_PLAN_SMC	PROFILE_VARIABLE_CD...	String	SMC	(no values)
PROFILE_BREAKDOWN_BC	PROFILE_VARIABLE_CD...	String	BC	(no values)
R_CENSURING	RightCensuring (Value>1) ...	Real	1.0	(no values)
L_CENSURING	LeftCensuring (Value<1) B...	Real	0.999	(no values)

Figure 52. Parameters used for Deployment

In IBM SPSS, you can find all of these parameters, but IBM Integration Bus (IIB) exposes only the RESOURCE_ID and the IS_1_RES_TRAIN flag in the IBM Predictive Maintenance and Quality release.

If there is a requirement to train one resource at a time, the resource id is passed along with the flag value.

The deployment type for both the health score and forecasted days to maintenance models is selected as **Model Refresh**, which supports scoring, building, and refresh features.

The following figure shows the deployment type for the health score model.

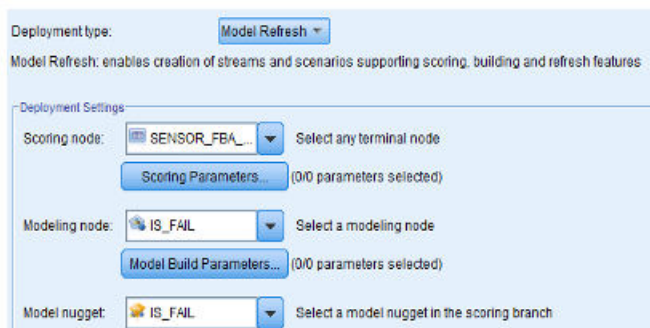


Figure 53. Refreshing the data for the scoring service

The following figure shows the deployment type for the forecasted days to maintenance model.

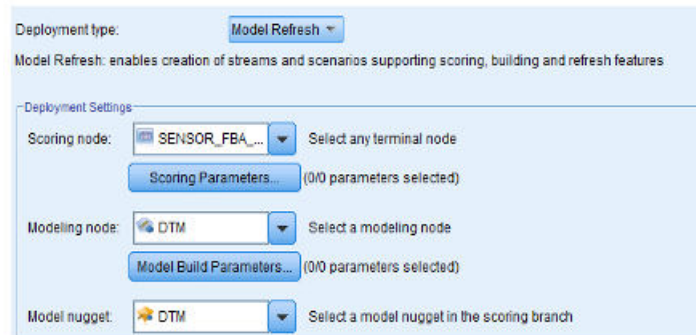


Figure 54. Refreshing the data for the scoring service

Recommendations

The Feature-based predictive model provides recommendations for each asset.

Feature-based analytics recommendations are produced by using the real-time mode of invocation. In the invocation mode, the stream is developed by using the ADM, which internally calls the health score and forecasted days to maintenance models, and a feature-based analytics service is configured for scoring services.

The service returns the health score (FHS), Forecasted Days to Maintenance (FFDM), deviation (DEV_FFDM), deviation percentage (DEV_FFDM_PCT) when compared to schedule maintenance (when available), and a recommended action (RECOMMENDATION) for each asset.

Depending on the health score, forecasted days to maintenance, and deviation percentage, the final recommendation is produced.

The following diagram shows an example of feature-based analytics recommendations.

Rule name	Allocate to	Insert rule	Sort	Remove
1 No Forecast Available	4001			
2 Urgent Inspection	FBA101			
3 Need Monitoring	FBA102			
4 Within Limits	FBA103			
5 No Schedule Available	4002			
6 Maintenance as Scheduled (0 to 10)	3001			
7 Maintenance as Scheduled (-10 to 0)	3002			
8 Prognose Maintenance (-25 to -10)	2001			
9 Prognose Maintenance (-30 to -25)	2002			
10 Prognose Maintenance (-75 to -50)	2003			

No Forecast Available

FHS IS NULL

OR FHS = -1

FFDM IS NULL

OR FFDM = 9999

Urgent Inspection

FHS BETWEEN 0.0 and 0.4

OR FFDM BETWEEN -9998.0 and 3.0

Figure 55. Recommendation settings for the Feature-based predictive model

The Integrated predictive model

The Integrated predictive model produces a predicted health score and days to maintenance for each asset or process at a site. The health score is used to determine an asset's performance.

The health score determines how likely an asset is to fail. This health score model can continuously monitor machine or asset health and predict potential machine faults in real time. It uses historical defect and maintenance data, along with

results of the other Analytics models, to determine the integrated health score and forecasted days to maintenance of an asset. The integrated model can also be used to predict the future health of an asset.

Input data for training

Integrated health score and days to maintenance training requires a combination of input data.

The following list describes the input data that the integrated model requires.

- sensor health score
- maintenance health score and predicted days to maintenance
- feature-based maintenance health score and predicted days to maintenance
- user-planned or user-scheduled days to maintenance, if available

Minimum data requirements

The minimum data that integrated analytics requires is three failure and three non-failure events per resource, or per resource sub type for sub type level models, with optimal data quality for other parameters.

The input data can be at event grain, day grain, or lifetime grain. For day grain data, any KPI date with even a single failure event in that day is considered a failure date. It is recommended to use noise treatments, as the training event uses data too. For event grain data, users can flag events before failure as fail events for realistic modeling.

Resources that do not satisfy the minimum data requirements are not trained, and they are logged in the Training Eligibility report as 0 against their eligibility to be trained. Any attempt to score these resources returns a health score of -1 or a predicted days to maintenance of -9999. These results are identified and dropped by IBM Integration Bus (IIB), and they are not available in databases for downstream applications.

Note: The data requirements are only an implemented bare minimum, and they do not guarantee an optimally trained model.

For resources that satisfy the bare minimum data requirements, IBM SPSS can sometimes give errors or fail some of the ensemble models. The errors or failures occur when the data quality is not good, or for other reasons. Under such conditions, the data volume and quality must be improved. As a last resort, you can remove the failing models from the expert modeler.

Resource sub type level modeling

Sub type level modeling is sometimes needed to supplement resource level modeling.

In many cases, equipment failures occur after a long duration, and they are replaced or refurbished after the failure. Because IBM Predictive Maintenance and Quality 2.0 Integrated Analytics is trained per resource, there are no trainable resources for these failures.

Users can have a custom segmentation model that will segment resources, even within a type, based on similar characteristics, and use that segment information within the model. Predictive Maintenance and Quality 2.5 supports segmentation at a resource sub type level.

During scoring, the health score and forecasted days to maintenance come from the most specific model that was trained. For example, for a resource, if the resource level model was not trained, then the resource sub type model is used for scoring. If the resource level model was trained, then that model is used for health scores and forecasted days to maintenance.

Training job

The `IBMPMQ_INTEGRATED_FEATURE_BASED_ANALYTICS` job calls the data preparation stream, followed by training and refreshing the health score and forecasted days to maintenance models.

The following diagram shows the `IBMPMQ_INTEGRATED_FEATURE_BASED_ANALYTICS` job.

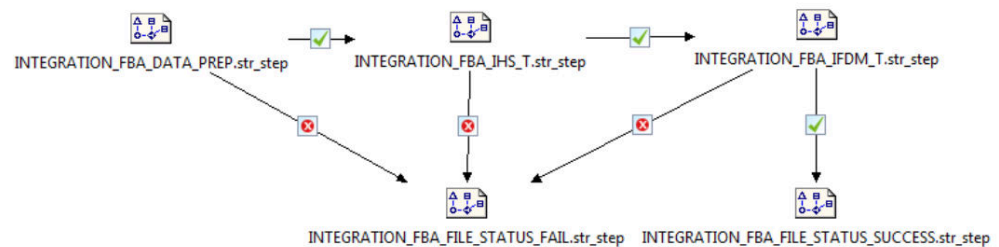


Figure 56. The `IBMPMQ_INTEGRATED_FEATURE_BASED_ANALYTICS` job

At the end of the job, the file `Training_IHS_in.csv` is returned to the `\root\integrationin` directory. The file contains two fields, System Timestamp and Status, which are read by the IBM Integration Bus (IIS).

If there is failure at any step in the job, the file contains FAIL as the status. When all steps are completed successfully, the file contains SUCCESS as the status.

Data preparation

Data preparation for the Integrated predictive model occurs during execution of the `INTEGRATION_FBA_DATA_PREP.str` stream.

The Integrated Analytics model produces a final predicted health score and forecasted days to maintenance for each asset at a site. The health score is used to determine an asset's performance.

The health score determines how likely an asset is to fail (inversely proportional), and the forecasted days to maintenance determines optimal days to the next maintenance. Together, these models can continuously monitor machine or asset health, and predict potential machine faults and/or optimal maintenance schedules in real time.

The `INTEGRATION_FBA_DATA_PREP.str` stream is described in the following table.

Table 27. The INTEGRATION_FBA_DATA_PREP.str stream

Stream name	Purpose	Input	Output
INTEGRATION_FBA_DATA_PREP.str	A Data preparation stream extracts the data from IBM Predictive Maintenance and Quality tables, and prepares the data to be used in the modeling. The eligible data is exported to a csv file for the modeling.	The input data source contains sensor, maintenance, and feature-based analytics health scores and forecasted days to maintenance information of machines, in addition to the scheduled maintenance details.	A list of machines for which there is enough data and that are eligible for training to identify the patterns. Transformed data input to modeling for the eligible resources.

This stream prepares the data for analysis of the health score and forecasted days to maintenance, based on the output of the other analytics models, namely Sensor (HS), Sensor FBA (FHS and FFDM), Maintenance (MHS and FDM), and scheduled days to maintenance (SDM) when available. In addition, to train the model to identify failure patterns, enough failure data must be available.

Machines that do not have sufficient failure data are not eligible for further modeling. Machine IDs are logged in the Training_Eligibility_IntegrationFBA_Report.csv file. In this file, resources are indicated either with 1 (eligible) or 0 (not eligible).

Orchestration rules

Integrated Analytics takes input from Sensor, Feature-Based, and Maintenance Analytics.

In IBM Predictive Maintenance and Quality, the Integration Analytics model is called after Feature-Based Analytics, through the same orchestration rule configured for the event type FEATURE.

The Integration Analytics model is invoked through the Scoring adapter, which invokes the model through the Representational State Transfer (REST) interface. The Scoring adapter is exposed by Analytics Solutions Foundation.

IBM SPSS exposes a single service to invoke the scoring model and the ADM service. The service returns the integrated health score (IHS), Integrated Forecasted Days to Maintenance (IFDM), deviation (DEV_IFDM), deviation percentage (DEV_IFDM_PCT) when compared to schedule maintenance (when available), and a recommended action (RECOMMENDATION) for each asset. In Predictive Maintenance and Quality, the recommendation is processed as an event.

SPSS results are stored in the EVENT_PROFILE and RESOURCE_PROFILE tables.

Predictive modeling

The Integrated predictive model uses different streams for health score modeling and for forecasted days to maintenance modeling.

Health score modeling

The following table describes the INTEGRATION_FBA_IHS_T.str stream that is used for health score modeling.

Table 28. The INTEGRATION_FBA_IHS_T.str stream

Stream name	Purpose	Input	Target	Output
INTEGRATION_FBA_IHS_T.str	Predicts the failure of equipment based on Sensor, Feature-based, and Maintenance Analytics model-based health scores, in addition to scheduled and forecasted maintenance details. Trains the models, and refreshes them for the scoring service.	The input data source contains the health scores from Sensor (HS), Feature-based (FHS), and Maintenance (MHS) models, in addition to the scheduled (SDM) and forecasted maintenance (FDM from Maintenance model and FFDM from the Feature-based model) results	IS_FAIL	Integrated health score of the equipment

Depending on the input data, you might have to consider a different approach for the integrated health score modeling. In addition, the concept of Splits is introduced at a tenant ID, location ID, resource sub type, and resource ID level in the Type Node, because the trained model must be unique for each combination of tenant, location, and resource.

The value of the health score of an asset is 0 - 1. The higher the value of the health scores, the better the health of the asset. If the input data model and structure is modified, the health score model must be retrained on the new data.

The health score model is based on the IBM SPSS Modeler auto classification model's confidence. Alternatively, raw and adjusted raw propensity scores can be used to generate such scores. In the model node, there are various modeling technique-specific settings. These settings can be configured based on the requirements and the available data. Similarly, the data in this case is not balanced. Depending on the data and requirements, balancing might give better results.

Note: Not all of the models support propensity score outputs, especially when splits are enabled.

Forecasted days to maintenance modeling

The following table describes the INTEGRATION_FBA_IFDM_T.str stream that is used for forecasted days to maintenance modeling.

Table 29. The INTEGRATION_FBA_IFDM_T.str stream

Stream name	Purpose	Input	Target	Output
INTEGRATION_FBA_IFDM_T.STR	Predicts forecasted days to next maintenance based on Sensor, Feature-based, and Maintenance Analytics model-based health scores, in addition to scheduled and forecasted maintenance details. Trains the models, and refreshes them for the scoring service.	The input data source contains the health scores from Sensor (HS), Feature-based (FHS), and Maintenance (MHS) models, in addition to the scheduled (SDM) and forecasted maintenance (FDM from Maintenance model and FFDM from the Feature-based model) results	DTM (Days to maintenance based on the maintenance or failure records)	Integrated forecasted days to the next maintenance for the equipment

Depending on the input data, you might have to consider a different approach for the integrated forecasted days to maintenance modeling. In addition, the concept of Splits is introduced at a tenant ID, location ID, resource sub type, and resource ID level in the Type Node, because the trained model must be unique for each combination of tenant, location, and resource.

The value of the forecasted days to maintenance determines the optimal maintenance period. The lower the value, the earlier the maintenance is recommended.

The forecasted days to maintenance are based on the SPSS Modeler auto numeric model. In the model node, there are various modeling technique-specific settings. These settings can be configured based on the requirements and available data. Similarly, the data in this case is not balanced. Depending on the data and requirements, balancing might give better results.

Deployment

The Integrated predictive model uses a data preparation stream that determines that the data for modeling is developed by using parameters that must also be used during deployment.

Some parameters are configured in the downstream applications. If the parameter values are passed when the stream is executed, these values are used. Otherwise, default values are used. See the following figure.

Name	Long name	Storage	Value	Type
IS_1_RES_TRAIN	Train for a Single Resource	Integer	0	(no values)
RESOURCE_ID	Which Resource to train for...	Integer	461	(no values)
PROFILE_PLAN_AMC	PROFILE_VARIABLE_CD...	String	AMC	(no values)
PROFILE_PLAN_SMC	PROFILE_VARIABLE_CD...	String	SMC	(no values)
PROFILE_BREAKDOWN_BC	PROFILE_VARIABLE_CD...	String	BC	(no values)
R_CENSURING	RightCensuring (Value>1) ...	Real	1.0	(no values)
L_CENSURING	LeftCensuring (Value<1) B...	Real	0.999	(no values)

Figure 57. Parameters used for Deployment

In IBM SPSS, you can find all of these parameters, but IBM Integration Bus (IIB) exposes only the RESOURCE_ID and the IS_1_RES_TRAIN flag in the IBM Predictive Maintenance and Quality release.

If there is a requirement to train one resource at a time, the resource id is passed along with the flag value.

The deployment type for both the health score and forecasted days to maintenance models is selected as **Model Refresh**, which supports scoring, building, and refresh features.

The following figure shows the deployment type for the health score model.

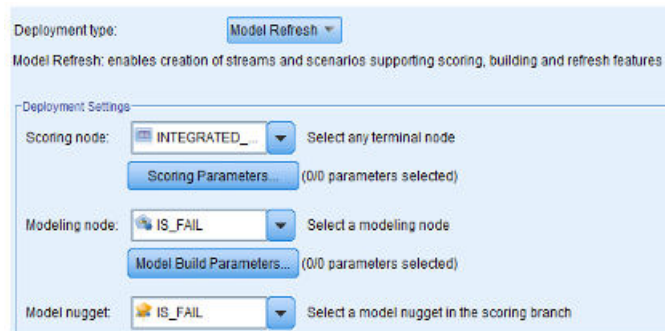


Figure 58. Refreshing the data for the scoring service

The following figure shows the deployment type for the forecasted days to maintenance model.

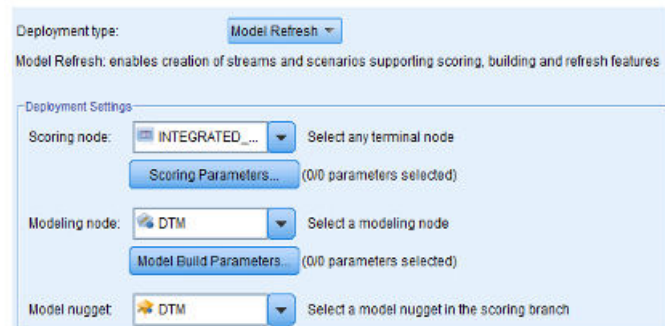


Figure 59. Refreshing the data for the scoring service

Recommendations

The Integrated predictive model provides final recommendations for each asset, based on the results of other analytics models on Sensor, Maintenance, and Feature-based data.

Integrated analytics recommendations are produced by using the real-time mode of invocation. In the invocation mode, the stream is developed by using the ADM, which internally calls the health score and forecasted days to maintenance models, and an INTEGRATED_FBA service is configured for scoring services.

The service returns the integrated health score (IHS), Integrated Forecasted Days to Maintenance (IFDM), deviation (DEV_IFDM), deviation percentage

(DEV_IFDM_PCT) when compared to schedule maintenance (when available), and a recommended action (RECOMMENDATION) for each asset.

Depending on the health score, forecasted days to maintenance, and deviation percentage, the final recommendation is produced.

The following diagram shows an example of Integrated analytics recommendations.

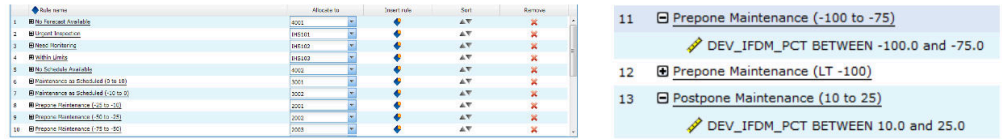


Figure 60. Recommendation settings for the Integrated predictive model

Chapter 9. Recommendations

When an asset or a process is scored and identified as having a high probability of failure, recommendations can be generated.

Define recommended actions by using rules in IBM Analytical Decision Management. Use IBM Analytical Decision Management to understand the drivers that are used to define the rules, and to determine what happens based on the scores received. For example, if a score breaches a threshold, what is the resulting action? You can automate alerts for recommended actions by integrating with other systems or by defining a routing rule to send emails. Depending on the manufacturing execution systems (MES) you use, the recommendation can be acted on automatically. You can also predict the success rate of the corrective action that is based on previous actions.

For information about using IBM Analytical Decision Management, see IBM Analytical Decision Management Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SS6A3P>).

When IBM Predictive Maintenance and Quality generates recommendations, for example, to inspect an asset, you can configure the system so that the recommendation results in a work order that is created by IBM Maximo. The work order is populated with the information needed to complete the task, for example, a device identifier and a location.

Two IBM Analytical Decision Management templates are provided with IBM Predictive Maintenance and Quality:

- Prioritize application template
- Combine application template

Prioritize application template

Use the prioritize application template when you have a good understanding of the predictive analytics scores, and the interaction between the predictive scores. You can use this template to prioritize your business objective that is based on, for example, profit maximization, or downtime minimization.

The template is stored in the following location: `/opt/IBM/SPSS/Deployment/5.0/Server/components/decision-management/Templates/PredictiveMaintenanceQuality.xml`

This template contains the following information that can be customized:

- Input source data: contains the health score and expected device life time data from the IBM SPSS Modeler stream output. Additionally, it contains the calculations such as Mean, Minimum, Maximum values for a particular resource for a specific time stamp.
- Defined rules: resource recommendations are given based on the rules defined. The recommended actions are classified as **Urgent inspection**, **Need monitoring**, or **Within limits**.
- Prioritization: you can define the optimization objective for the business, for example, “profit maximization” or “downtime or loss minimization”.

Combine application template

Use the combine application template to use existing rules along side new predictive scores. This is useful if there are many rules that you do not want to replace with new predictive scores straight away. You can define a precedence structure for these rules to enable them to co-exist.

The template is stored in the following location: `/opt/IBM/SPSS/Deployment/5.0/Server/components/decision-management/Templates/PredictiveMaintenance.xml`

This template contains the following information that can be customized:

- Input source data: contains the health score and expected device life time data from the IBM SPSS Modeler stream output. Additionally, it contains calculations such as Mean, Minimum, Maximum values for a particular resource for a particular time stamp.
- Defined rules: logic-based business rules with appropriate risk points.
- Combine: specify the precedence order when the actions from the business rules and the model do not match.

Preventing scoring for incoming events

You can prevent scoring from being performed by IBM SPSS for incoming events. If you require IBM Maximo work order creation, you must not prevent scoring. By default, scoring is enabled (`SPSSTRIGGER` is set to `TRUE`).

Procedure

1. In the IBM WebSphere MQ Explorer, expand the **Brokers** node, the **MB8Broker** node, the **PMQ1** node, the **PMQEventLoad** node, right-click the **StdEventLoad** item, and click **Properties**.
2. Click **User Defined Properties**.
3. Set the `SPSSTRIGGER` property to `FALSE`. To re-enable scoring, set the `SPSSTRIGGER` property to `TRUE`.

Disabling work order creation

If IBM Maximo is not integrated with your IBM Predictive Maintenance and Quality installation, or if you want to disable work order creation, do the following steps:

Procedure

1. In the IBM WebSphere MQ Explorer, go to **Brokers > MB8Broker > PMQ1**. Right-click the **PMQIntegration** node, and click **Properties**.
2. Click **User Defined Properties**.
3. Set the `MaximoTRIGGER` value to `FALSE`. To re-enable work order creation, set the `MaximoTRIGGER` property to `TRUE`. By default, the `MaximoTRIGGER` property is set to `FALSE`.

Chapter 10. Reports and dashboards

You can customize and extend the reports and dashboards that are supplied with IBM Predictive Maintenance and Quality. You can also design your own reports and dashboards and add them to the menu.

You can use IBM Cognos Report Studio to create scorecards and reports. Before you run the reports, familiarize yourself with the behavior of reports in Report Studio. For example, a star next to a prompt indicates that it is required. For information about how to use Report Studio, see *IBM Cognos Report Studio User Guide*. You can obtain this user guide from IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/>).

Tip: Images that are needed for custom and extended reports must be placed in the `/opt/ibm/cognos/c10_64/webcontent/Images` folder on the Business Intelligence node. The image URL property value in Report Studio for images placed in this directory is the name of the file.

You can modify the data model for these reports by using IBM Cognos Framework Manager. For more information, see Appendix C, “IBM Cognos Framework Manager model description,” on page 197.

The following table describes the reports available from the Site Overview Dashboard.

Table 30. Reports available from the Site Overview Dashboard

Reports	Description
Overview	Provides a high-level summary of the health of all of your assets at all sites, it shows the key performance indicators (KPIs) with the greatest impact. You can change the detail that is displayed by selecting items from the list boxes. For example, you can change the date and the equipment type.
Top 10 Contributors	Identifies the equipment, locations, and operators responsible for the most failures.
KPI Trending	You can select multiple key performance indicators (KPIs) to be plotted side-by-side in a line chart. You can identify correlations between the KPIs and see whether there is any lagging behavior. For example, if there is a spike in one KPI, how long does it take to impact the other KPIs?
Actual vs Plan	You can monitor how closely the metrics track against the plan. Variances are highlighted.
Equipment Listing	The health score for a site is derived from the lower-level scores from each piece of equipment in the site. This report shows you all the pieces of equipment on the site and the health scores and relevant KPIs for that equipment.
Equipment Outliers	Lists the equipment (or assets) that are performing outside of allowable limits. The measures that are shown differ depending on the equipment, but examples are operating temperature, lateral strain, hydraulic pressure, average value, last value, and control limits.

Table 30. Reports available from the Site Overview Dashboard (continued)

Reports	Description
List Of Recommended Actions	A summary of all recommended actions for each piece of equipment, for the health score measurement.

The following table describes the reports that are available from the Equipment dashboard.

Table 31. Reports available from the Equipment dashboard

Reports	Description
Equipment Profile	A detailed report that shows everything that is known about a piece of equipment: how it is performing today and how it performed in the past.
Equipment Control Chart	Shows the upper and lower control limits and the average limits for selected measures.
Equipment Run Chart	Shows the measures for a particular piece of equipment.
Equipment Outliers	Shows detailed measures for a piece of equipment that shows anomalies.
Event Type History	Lists the events for a device.

The following table describes the reports that are available from the Product Quality Dashboard.

Table 32. Reports available from the Product Quality Dashboard

Reports	Description
Defect Analysis	Shows product defects and inspection rates.
Inspection Rate Analysis	Examines the relationship between inspections and defects over time to find the optimal rate of inspection.
Material Usage By Process	Provides an overview of material usage in the production processes.

Site Overview Dashboard

The Site Overview Dashboard provides a high-level summary of the health of all of your assets at all sites. It shows the key performance indicators (KPIs) with the greatest impact. It contains the Site Summary Report, the Health Score Trend Bar Chart, the Health Score Contributors Pie Chart, and the Incident and Recommendation Analysis Bar Chart.

You can use the following prompt filters in the dashboard. The filters are applied to all reports and charts on the dashboard:

- From Date
- To Date
- Location
- Resource sub type

Site Summary Report

The following table describes the measures for the Site Summary Report.

Table 33. Site summary measures

Measures	Description
Health Score	The evaluation of the health of a resource that is based on predictive models.
Resource Count	The number of resources.
Incident Count	Counts the number of failures that are recorded by resources.
Alarm Count	Counts the number of alarms that are generated by resources.
Recommendation Count	A recommendation is generated by the predictive model when a resource approaches a failure. This measure counts the number of recommendations generated.
MTTR (Mean time to repair)	Average time, for example, in hours between the occurrence of an incident and its resolution, which is calculated by using the following calculation: Repair Time / Repair Count
MTBF (Mean time between failures)	The average time between equipment failures over a period. For example, the average time a device functions before it fails. It is the reliability rating that indicates the expected failure rate of equipment. It is calculated by using the following calculation: Operating Hours Delta / Incident Count

The measure source is the resource_kpi table. The measures are displayed by location.

Health Score Trend Bar Chart

The Health Score Trend Bar Chart uses the Health score measure. The measure source is the resource_kpi table. The measure is displayed by date.

Health Score Contributors Pie Chart

The Health Score Contributors Pie Chart uses the Health score measure. The measure source is the resource_kpi table. The measure is displayed by resource.

Incident and Recommendation Analysis Bar Chart

You can use this report to analyze incidents and recommendations.

You can access the Drill Through - Incident and Recommendation Event List drill through report from the Incident and Recommendation Analysis Bar Chart:

Note: The drill through reports are stored in the **Drill Through Reports** folder. The reports in this folder are intended to be run from the main report with which they are associated. Do not run the drill through reports on their own.

The following table describes the measures for the **Incident and recommendation analysis** bar chart. The measure source is the resource_kpi table. The measures are displayed by date.

Table 34. Incident analysis bar chart

Measures	Description
Recommendation Count	A recommendation is generated by the predictive model when a resource approaches a failure. This measure counts the number of recommendations that are generated.
Incident Count	Counts the number of failures that are recorded by resources.

Recommendation and Incident Count Accessibility List

This chart provides the same information as the Incident and Recommendation Analysis Bar Chart in an accessible format.

The Recommendation and Incident Count Accessibility List contains the following drill through reports:

Drill Through - Incident Event List

This report shows the incident event list in tabular form.

Drill Through - Recommendation Event List

This report shows the recommendation event list in tabular form.

Note: The drill through reports are stored in the **Drill Through Reports** folder. The reports in this folder are intended to be run from the main report with which they are associated. Do not run the drill through reports on their own.

Top 10 Contributors dashboard

The Top 10 Contributors dashboard identifies the top 10 equipment, locations, and operators responsible for the most failures.

The following table indicates which dimension is used to display the Actual Value measure in each report.

Table 35. Dimensions that display the Actual Value measure in Top 10 Contributors reports

Report	Dimension
Top 10 Contributors by Resource	Resource
Top 10 Contributors by Location	Location
Top 10 Contributors by Organization	Group Dimension

The Actual Value measure is the aggregated value of an actual event measurement. Depending on the aggregation type of the Profile Variable, it can be calculated by using the following calculation: $\text{sum (Actual Value) / sum (Measure Count)}$ or $\text{sum (Actual Value)}$

The measure source is the resource_kpi table. The measures are displayed by the Location hierarchy.

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type

- Profile variable

KPI trending report

Users can select multiple key performance indicators (KPIs) to be plotted side-by-side in a line chart. You can identify correlations between the KPIs and see whether there is any lagging behavior. For example, if there is a spike in one KPI, how long does it take to impact the other KPIs?

The KPI trending report uses the Actual Value measure, which is the aggregated value of an actual event measurement. Depending on the aggregation type of the Profile Variable, it can be calculated by using the following calculation: $\text{sum}(\text{Actual Value}) / \text{sum}(\text{Measure Count})$ or $\text{sum}(\text{Actual Value})$. The measure is displayed by the Calendar hierarchy, and the measure source is the `resource_kpi` table.

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type
- Profile variable

Actual vs plan report

This report monitors how closely the metrics are tracking against the plan. Variances are highlighted when a metric is off-track.

The following table describes the measures, and the measure source for the **Actual vs plan** report.

Table 36. Measures and the measure source in the Actual vs plan report

Measures	Measure description	Measure source
Plan Last Value	The last recorded planned value for the resource. "Planned" is determined by the Value Type	resource_profile table
Actual Last Value	The last recorded actual value for the resource. "Actual" is determined by the Value Type	Resource Profile
Variance	Plan value - Actual Value	Report calculation

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type
- Profile variable

Equipment listing report

The health score for a site is derived from the lower-level scores from each piece of equipment in the site. Use this report to view all the pieces of equipment on the site and the health scores and relevant key performance indicators (KPIs) for that equipment.

The following table describes the measures for the Equipment listing report. The measure source is the resource_kpi table. The measures are displayed by the Resource hierarchy.

Table 37. Measures in the Equipment listing report

Measures	Description
Health Score	The evaluation of the health of a resource based on predictive models.
Work Order Count	This counts the number of work orders issued. A Work Order is a separate event type from resource measurements.
Incident Count	This counts the number of failures recorded by resources.
Recommendation Count	A recommendation is issued by the predictive model when a resource approaches a failure. This measure counts the number of recommendations that have been issued.
MTBF (Mean time between failures)	The average time between equipment failures over a given period, for example, the average time a device will function before failing. It is the reliability rating indicating the expected failure rate of equipment. It is calculated by using the following calculation: Operating Hours Delta / Incident Count.
MTTR (Mean time to repair)	Average time (for example, in hours) between the occurrence of an incident and its resolution, calculated by using the following calculation: Repair Time / Repair Count

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type

Outliers report

This reports lists the equipment or assets that are performing outside of allowable limits.

The following table provides the measure details for the Outliers report.

Table 38. Measures in the Outliers report

Measures	Measure description	Measure source
Life to Date Average	The daily average measurement for the resource.	Resource Profile
Upper Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] + [Sigma Level] * [Life-to-Date Standard Deviation]	Report calculation

Table 38. Measures in the Outliers report (continued)

Measures	Measure description	Measure source
Lower Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] - [Sigma Level] * [Life to Date Standard Deviation]	Report calculation
Last Value	The most recent recorded measurement for this resource.	Resource Profile

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type
- Sigma level

Recommended actions report

This report summarizes all recommended actions for each piece of equipment.

The Recommended actions report uses the Health Score measure, which is the evaluation of the health of a resource that is based on predictive models. The measure is displayed by the Event Observation Resource hierarchy, and the measure source is the event table.

The following prompt filters are applied to this report:

- Date
- Location
- Resource sub type

Equipment dashboard

The Equipment dashboard gives you access to the Equipment profile report, the Equipment control chart, the Equipment run chart, the Equipment outliers chart, and Event type history chart.

Equipment profile report

The Equipment profile report is a detailed report that shows everything that is known about a piece of equipment: how it is performing today and how it performed in the past.

The following table provides the measure description for the **Equipment profile** report. The measure source is the resource_profile table. The measures are displayed by the Profile Variable hierarchy.

Table 39. Measures in the Equipment profile report

Measures	Measure description
Period Minimum	The lowest actual reading that is recorded for the resource measurement this period.
Period Maximum	The highest actual reading that is recorded for the resource measurement this period.

Table 39. Measures in the Equipment profile report (continued)

Measures	Measure description
Period Average	The daily average measurement for the resource.
Last Value	The most recent recorded measurement for this resource.
Period Total	The total actual reading that is recorded for the resource measurement this period.

The following prompt filters are applied to this report:

- Resource sub type
- Resource name
- Resource code
- Location
- Event code

Equipment control chart

The Equipment control chart shows the upper and lower control limits and the average limits for selected measures.

The following table provides the measure details for the **Equipment control chart** report.

Table 40. Measures in the Equipment control chart

Measures	Measure description	Measure source
Life to Date Average	This is an average measurement that is calculated over the life of the resource.	resource_profile table
Upper Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] + [Sigma Level] * [Life-to-Date Standard Deviation]	Report calculation
Lower Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] - [Sigma Level] * [Life-to-Date Standard Deviation]	Report calculation
Measurement	The actual value recorded on an event.	event table

The following prompt filters are applied to this report:

- Resource sub type
- Resource name
- Resource code
- Location
- Event code
- Calendar date
- Start time
- End time
- Measurement type

- Profile variable
- Sigma level

Equipment run chart

The equipment run chart shows the measures for a particular piece of equipment.

The Equipment run chart uses the Measurement measure, which is the actual value that is recorded on an event. The measure source is the event table, and the measure is displayed by the event time hierarchy.

The following prompt filters are applied to this report:

- Resource sub type
- Resource name
- Resource code
- Location
- Event code
- Calendar date
- Start time
- End time
- Measurement type

Equipment outliers

The equipment outliers report shows detailed measures for a piece of equipment that shows anomalies.

The following table describes the measures for the Equipment outliers report. The measures are displayed by the Profile Variable hierarchy.

Table 41. Measures in the Equipment outliers report

Measures	Measure description	Measure source
Life to Date Average	This is an average measurement that is calculated over the life of the resource.	resource_profile
Upper Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] + [Sigma Level] * [Life-to-Date Standard Deviation]	Report calculation
Lower Control Limit	This is calculated by using the following calculation: [Life-to-Date Average] - [Sigma Level] * [Life-to-Date Standard Deviation]	Report calculation
Last Value	The most-recent recorded measurement for this resource.	resource_profile

The following prompt filters are applied to this report:

- Resource sub type
- Resource name
- Resource code

- Location
- Event code

Event type history report

The event type history report lists the events for a device.

The Event type history report uses the Measurement measure, which is the actual value that is recorded on an event. The measure source is the event table, and the measure is displayed by the Event Time, Measurement Type and Event Observation.

The following prompt filters are applied to this report:

- Resource sub type
- Resource name
- Resource code
- Location
- Event code
- Calendar date
- Event type

Product quality dashboard

The Product quality dashboard highlights areas of the production process that are affected by defects, and enables you to see if any relationships exist between the rate of inspections, and the rate of defects.

Defect analysis dashboard

The Defect analysis dashboard provides an overview of product defects and inspection rates. The dashboard is made up of a number of reports that analyze defects by event code, location, and production batch.

Defect summary

This report analyzes product defects and inspection rates.

The following table describes the measures for the **Defect summary** report. The measure source is the process_kpi table. The measures are displayed by the Product hierarchy.

Table 42. Measures in the Defect summary report

Measures	Measure Description
Defect Count	The number of defects that are reported.
Quantity Produced	The quantity that is produced.
Defect Rate	Defect Count divided by Quantity Produced.
Quantity Planned	The quantity that is expected to be produced.
Defect Target	The acceptable number of defects.
Test Failure Rate	Test Failures divided by Number of Tests.
Target Defect Rate	Defect Target divided by Quantity Planned.
Inspection Time	The amount of time that is spent inspecting the product.

Table 42. Measures in the Defect summary report (continued)

Measures	Measure Description
Assembly Time	The amount of time that is spent producing the product.
Inspection Time Rate	Inspection Time divided by the Assembly Time.
Inspection Count	The number of inspections performed.
Inspection Rate	Inspection Count divided by Quantity Produced.
Average Assembly Time	.Assembly Time divided by Quantity Produced.

The following prompt filters are applied to this report:

- Process hierarchy
- Calendar from/to date

Defects by event code

This pie chart shows product defects by event code, also known as failure code.

The Defects by event code report uses the Actual Value measure, which is the aggregated value of an actual event measurement. Depending on the aggregation type of the Profile Variable, it can be calculated by using the following calculation: $\text{sum (Actual Value)} / \text{sum (Measure Count)}$ or $\text{sum (Actual Value)}$

The measure is displayed by the Event Code hierarchy, and the measure source is the process_kpi table.

The following prompt filters are applied to this report:

- Process hierarchy
- Calendar from/to date

Defects by location

This pie chart shows product defects by location.

The Defects by location report uses the Actual Value measure, which is the aggregated value of an actual event measurement. Depending on the aggregation type of the Profile Variable, it can be calculated by using the following calculation: $\text{sum (Actual Value)} / \text{sum (Measure Count)}$ or $\text{sum (Actual Value)}$

The measure is displayed by the Location hierarchy, and the measure source is the process_kpi table.

The following prompt filters are applied to this report:

- Process hierarchy
- Calendar from/to date

Defects by production batch

This pie chart shows product defects by production batch.

The Defects by production batch report uses the Actual Value measure, which is the aggregated value of an actual event measurement. Depending on the aggregation type of the Profile Variable, it can be calculated by using the following calculation:

$\text{sum (Actual Value) / sum (Measure Count) or sum (Actual Value)}$

The measure is displayed by the Production batch hierarchy, and the measure source is the process_kpi table.

The following prompt filters are applied to this report:

- Process hierarchy
- Calendar from/to date

Inspection rate analysis

This report examines the relationship between inspections and defects over time in order to find the optimal rate of inspection.

It is made up of the Defect Summary report, the Defect plan vs actual bar chart, and the Defect rate vs inspection rate line chart.

Defect plan vs actual report

The following table provides the measure details for the Defect plan vs actual report. The measure is displayed by the Product hierarchy, and the measure source is the process_kpi table.

Table 43. Measures in the Defect plan vs actual report

Measures	Measure description
Defect Rate	Defect Count divided by Qty Produced.
Target Defect Rate	The rate of the Defect Target divided by Quantity Planned measure.

The following prompt filters are applied to this report:

- Process hierarchy
- Calendar from and to date

Defect rate vs inspection rate line chart

The following table provides the measure details for the Defect rate vs inspection rate line chart. The measure is displayed by the Calendar hierarchy, and the measure source is the process_kpi table.

Table 44. Measures in the Defect rate vs inspection rate line chart

Measures	Measure description
Defect Rate	Defect Count divided by Quantity Produced.
Inspection Rate	Inspection Count divided by Quantity Produced.

The following prompt filters are applied to this report:

- Process hierarchy

- Calendar from/to date

Material usage by process crosstab

This report provides an overview of material usage in the production processes.

This report includes the Defect Summary report.

This report uses the Period Measure Count, which is the number of measurements that are taken in one period. By default, a period is one day. The measure is displayed by the Material by Type, Supplier, and Batches by Product hierarchies, and the measure source is the `material_profile` table.

The Process hierarchy prompt filter is applied to this report.

Audit Report

The Audit Report shows the count of rows in the major master data tables.

Note: The Asset count is shown in the Audit Report.

The Audit Report contains the following drill through reports:

Drill Through - Resource List

Lists the resources for a resource type.

Example

For example, the Audit Report shows the count for the Asset resource type. Click on this count number to open the Drill Through - Resource List report that lists all of the assets.

Drill Through - Profile Variables

Lists all measures and key performance indicators that are being tracked in daily profiles and historical snapshots.

Drill Through - Process List

Lists all production processes.

Drill Through - Material List

Lists materials that are used in the production process.

Drill Through - Production Batch List

Lists production batches for a product.

Drill Through - Measurement Type List

Lists measurement types. For each measurement type, the report shows unit of measure and aggregation type.

Note: The drill through reports are stored in the **Drill Through Reports** folder. The reports in this folder are intended to be run from the main report with which they are associated. Do not run the drill through reports on their own.

The following table describes the measures in the Audit Report. The measure source is the report calculation.

Table 45. Measures in the Audit Report

Measures	Measure Description	Hierarchies
Count of Resources by Type	A count of rows in the dimension	Resource by Type

Table 45. Measures in the Audit Report (continued)

Measures	Measure Description	Hierarchies
Count of Materials by Type	A count of rows in the dimension	Material by Type
Count of Profile Variables	A count of rows in the dimension	Profile Variable
Count of Measurement Types	A count of measurement types in the dimension	Measurement Type
Count of Processes	A count of rows in the dimension	Process
Count of Production Batches by Product	A count of rows in the dimension	Batches by Product

Material usage by production batch

This report provides an overview of material usage by production batch. By correlating production batches with defects to material usage by production batch, you can begin to trace the impact of defective materials.

The material usage by production batch report uses the Period Measure Count, which is the number of measurements that are taken in one period. By default, a period is a day. The measure is displayed by the following hierarchies:

- Batches by Product
- Supplier
- Material by Type

The measure source is the `material_profile` table.

The following prompt filters are applied to this report:

- Process hierarchy
- Event code

Maintenance Overview Dashboard

The Maintenance Overview Dashboard provides insights by using existing maintenance data and can include sensor data when your organization's data matures. The Maintenance Overview Dashboard also gives you insight into stable life and rapid wear scenarios.

This report shows the sensor health score, maintenance health score, the integrated health score, and the feature based health score by location and resource for the last current day in the record. The sensor health score is a near real-time value calculated from the sensor reading. The maintenance health score is calculated from the maintenance logs. The sensor health score and the maintenance health score are combined to give the integrated health score.

Recommendations are shown for each health score value to help users take the necessary actions. The prompt filters forecasted days to next maintenance, scheduled days to next maintenance, feature based forecasted days to next maintenance, and integrated forecasted days to next maintenance, along with their deviations, are shown in the report, which help users to proceed with or postpone the scheduled maintenance cycle.

You can set the following prompt filters in this chart:

- Location
- Health Score
- Recommendation
- Absolute Deviation % (Maintenance)
- Absolute Deviation % (Feature based)
- Absolute Deviation % (Integrated)
- Forecasted Days to Next Maintenance
- Scheduled Days to Next Maintenance
- Feature Based Forecasted Days to Next Maintenance
- Integrated Forecasted Days to Next Maintenance
- Event Code
- Resource

The following columns are reported in this chart.

Table 46. Columns in the Maintenance Overview Dashboard

Column	Description
Location	Location of the resource.
Resource Sub Type	Sub type of the resource.
Resource	Identifies the resource.
Serial No	Serial number of the resource.
Health Score	The sensor health score, maintenance health score, and the integrated health score are values between 0.00 and 1.00. The higher the score, the better the performance of the resource.
Days To Next Forecasted & Scheduled Maintenance	The forecast number of days to next maintenance and number of days to next scheduled maintenance. The maximum positive deviation, minimum positive deviation, maximum negative deviation, and minimum negative deviation are also indicated.
Forecast - Schedule Deviation	The difference between the forecast days and the scheduled days.
Recommendation	The recommended action as indicated by the health scores.
Profile Variable CD	Profile Variable codes used in the Maintenance dashboard. The possible values are MHS, IHS, SHS, and FHS.

The Maintenance Overview Dashboard is designed on Materialized Query Tables (MQT) to improve the report response time. The MQTs that are created in the IBMPMQ database are:

- MAINTENANCE_PROFILE
- MAINTENANCE_EVENT

When the underlying tables are loaded, these tables must be refreshed. The crontab job is used to refresh these tables. The job runs ON scheduled basis on a daily basis, and it can be configured according to customer needs. The job runs every midnight to refresh these MQTs.

To see the data in the report immediately, refresh the MQTs manually by running the corntab script manually or by running the REFRESH command against these tables. The REFRESH command syntax is

```
REFRESH TABLE SCHMENAME.MQT_TABLENAME
```

For example, REFRESH TABLE PMQSCH.MAINTENANCE_PROFILE.

For more information about the corntab script, see the IBM Predictive Maintenance and Quality *Installation Guide*.

Maintenance Advance Sorting

Click **Advance Sorting** to drill through to the Maintenance Advance Sorting report. This report displays the same measures as the main report in a tabular format. You can sort on a column by clicking the column header. The prompt values from the main report are used in the Maintenance Advance Sorting report. You can change the prompt values in the Maintenance Advance Sorting report and run it with the new values.

Maintenance Health and Failure Detail Report

Click a resource in the **Resource** column to drill through to the Maintenance Health and Failure Detail Report for the resource.

The prompt values from the main report are used in this chart. You can change the following prompt filters in this chart and run it with the new values:

- From date
- To date
- Location
- Resource

You can include or exclude the following events:

- Breakdown maintenance
- Planned maintenance
- Forecasted maintenance
- Scheduled maintenance

Each event that you include appears as a bar on the chart. The bar indicates the date on which the event occurs. The health score, which is a value between zero and one, is indicated on the y axis. The x axis indicates the date of the health score. Health scores that occur before the current date are historical health scores. Health scores that occur after the current date are forecast health scores. The current health score is shown for the current date.

Click **Top N Failure Analysis** to drill through to the Top N Failure Analysis Report. For more information, see “TopN Failure Analysis Report” on page 171.

Note: It is possible that the location of a resource in the Maintenance Health and Failure Detail Report is different from the location of the same resource in the Top N Failure Analysis Report. If this happens, the **Location** field in the TopN Failure Analysis Report will be empty and you must select a location from the list and then run the report.

Statistical process control reports

The statistical process control (SPC) reports monitor the stability of a process. The charts in the reports show the data points in relation to the mean value, and upper and lower control limits.

SPC - Histogram

This bar chart is an overview of the frequency of an event or observation across a set of ranges or bins. The y axis shows the frequency. The x axis shows the bins. The height of the bar in a bin indicates the frequency of the event that falls into the range.

You can set the following prompt filters in this chart:

- From Date
- To Date
- Location
- Resource
- Event Type
- Measurement Type
- Event Code
- Number of Bins: Select **Number of Bins** to set the number of bins to display in the chart. The value that you select from the **User Selected Value** list is the number of bins that appears on the x axis.
- Bin Interval: Select **Bin Interval** to set the range for each bin. Enter the range in the **User Selected Value** field.
- Minimum: The minimum value for the bin range limit. Use this filter to set the lowest data point to be included in the data set.
- Maximum: The maximum value for the bin range limit. Use this filter to set the highest data point to be included in the data set.

The following measures are reported in the SPC Histogram chart.

Table 47. Measures in the SPC Histogram chart

Measure	Description
Frequency	Number of events that fall into a bin. The height of the bar indicates the frequency. Displayed on the y axis.
Bin Range	The bin interval. Displayed on the bins on the x axis.
Frequency of Bin Containing Mean Value	Frequency of the bin that contains the mean value of the events in the chart.
Count of Observations	Total number of events in the chart.
Mean	Mean value of the data in the chart.
Median	Median value of the data in the chart.
Minimum	Minimum value of the data in the chart.
Maximum	Maximum value of the data in the chart.
Range	The difference between the minimum and maximum values.
Standard Deviation	The standard deviation of the data in the chart.
Skewness	Indicates how symmetrical or asymmetrical the data is.
Kurtosis	Indicates if the data is peaked or flat, relative to a normal situation.

Table 47. Measures in the SPC Histogram chart (continued)

Measure	Description
Start Date	The date of the earliest event in the chart.
End Date	The date of the latest event in the chart.

The **Fitted Distribution** line shows the trend in the data.

Click **X Bar R/S Chart** to run the SPC - X Bar R/S Chart.

SPC - X Bar R/S Chart

The SPC - X Bar R/S chart shows variations in a process. You can use this chart to evaluate the stability of a process over a set of day ranges.

The SPC - X Bar chart shows how the average process changes over time. The median control limit is indicated by a dotted line. The solid lines on the chart indicate the upper and lower control limits. Data points that occur outside of the control limits indicate that the process is unstable.

The SPC - R/S chart shows how the average within a sub group changes over time. The SPC - R (range) chart is displayed when you enter a sub group value of 10 or less. The SPC - S (standard deviation) chart is displayed when you enter a sub group value that is greater than 10. The sub group size prompt controls the ranges that display on the x axis of both charts. For example, if you set the sub group prompt to 11 and the charts contain data from Jan 1 to Mar 9 (68 days), the x axis displays six ranges in 11-day increments. The seventh range contains a 2-day increment. The y axis on both charts indicates the control limit value.

The following prompts apply to this chart:

- From date
- To date
- Location
- Resource sub type
- Resource
- Measurement type
- Event code
- Resource code
- Profile variable type
- Sub group

Advanced KPI Trend Chart

This chart compares multiple key performance indicators (KPIs) across multiple resources. You can use this chart to analyze variations in a resource against a set of profiles. The main chart shows monthly data and you can drill down to a daily chart.

You can set the following prompt filters in this chart:

- From Date
- To Date
- Location

- Resource Sub Type
- Resource
- Profiles
- Event Code

Each chart displays data for one profile and all resources that you select in the prompt lists. By default, the chart displays all resources and all profiles but for clarity, select a few related profiles to analyze across a set of resources. Each data point in the chart represents one month of data for the profile. Click a data point or on the month on the x axis to see one month of data by day.

The following measures are reported in this chart.

Table 48. Measures in the Advanced KPI Trend Chart

Measure	Description
Actual Value	The value of the profile or measure for the resource for the month. Shown on the y axis.
Date	The year and month. Shown on the x axis. The month does not display if there is no data for the month.

QEWS quality dashboards

The QEWS quality dashboards show data for the QEWS quality inspection, warranty, and parametric use cases.

The quality dashboards provide an instant overview of parts and products with the help of stoplights. The following four factors are used to calculate stoplights:

- Return Code
- Severity
- Supplemental Alarms (also known as supplemental criteria)
- Business needs

The dashboards have a main report that shows the number of parts in each traffic light rule category, and a column that shows the overall distribution with respect to the product hierarchy level for a selected run date. The dashboards are linked to a detail history report, with a drill through link to show details of various threshold values with respect to each product of the drill through product hierarchy level. From the detail history report, a drill through link is provided to trigger the QEWS - Inspection chart report, the QEWSL - Warranty chart report, and the QEWSV - Parametric chart report.

Quality dashboard - Inspection

The Quality dashboard - Inspection report provides an overview of the state of products at a selected run date.

Prompt details

You can set the following prompt filter in this report:

- Run Date

Drill behavior

Drill up and drill down is enabled on the Product column. There is a drill through link from the Product column to the Quality Dashboard – Inspection Detail History report.

Quality dashboard - Inspection Detail History

The Quality dashboard - Inspection Detail History report provides details about the state of products and the various threshold values for a selected product category at a selected run date.

Prompt details

You can set the **Run Date** prompt filter in this report.

Drill behavior

There is a drill through link from the Product Code column to the QEWS – Inspection Chart report. The report is triggered from the Quality Dashboard - Parametric report.

Conditional formatting

Conditional formatting is applied to conditionally render the stoplight images for Distribution.

QEWS - Inspection Chart

The quality early warning system inspection chart reports the failure rates and values of evidence that the underlying failure rate process is unacceptably high.

You can report on a specific product type or a group of products. The analysis is based on data for a specified time period.

The chart shows performance of parts by vintage, where vintage is the day that the part was shipped. However, the analysis can be done for other vintages, such as the day of part manufacturing, or the day of part testing.

This chart is generated by IBM Predictive Maintenance and Quality daily. If the daily chart was not generated for the date that you select, the report is empty.

You can set the following prompt filters in this chart:

- Product Type
- Product Code
- Run Date

The chart heading contains the following information:

- Product code
- Last run date of the chart
- Period during which the product was shipped (start date and end date)
- Number of parts that were shipped over the period
- Number of parts that failed over the period
- Failure rate per 100 units over the period

Note: This chart is not an IBM Cognos Report Studio report so you cannot modify it in Report Studio.

Failure rate chart

This chart has a dual x axis that shows vintage number and Cumulative N_Tested. Vintage number is the day number that the part was shipped during the period. Cumulative N_Tested is the number of parts that were tested. The y axis shows the failure rate of the product per 100 units. A data point in the chart indicates the failure rate for a vintage number. The Acceptable Level is a horizontal line that indicates the acceptable failure rate.

Evidence chart

This chart has a dual x axis that shows vintage number and Cumulative N_Tested. Vintage number is the day number that the part was shipped during the period. Cumulative N_Tested is the number of parts that were tested. The y-axis shows the level of evidence that the underlying process failure rate is unacceptable, which is computed by using a weighted cumulative sum (CUSUM) formula.

The H Value is a horizontal line on the chart that shows the failure rate threshold value. The CUSUM values that are higher than H Value are displayed as triangles on the chart. The triangles indicate unacceptable process levels in the data. The vertical dotted line indicates the last time that the vintage number had an unacceptable failure rate. The forgiveness marker is the point in time when the process accumulated enough statistical evidence to suggest that its underlying failure rate was acceptable.

Summary list

The summary list header contains the same information as the chart heading. The summary list shows detailed information by vintage. It includes date, failure rate, total quantity that failed, and other data.

Quality dashboard - Warranty

The Quality dashboard - Warranty report provides an overview of the state of products at a selected run date.

Prompt details

You can set the following prompt filters in this report:

- Run Date
- Analysis Date

This prompt determines whether the Warranty start date is considered as the Date Sold, Date Manufactured, or Date Raw Materials Produced.

Drill behavior

Drill up and drill down is enabled on the Product column. There is a drill through link from the Product column to the Quality Dashboard – Warranty Detail History report.

Quality dashboard - Warranty Detail History

The Quality dashboard - Warranty Detail History report provides details about the state of products and the various threshold values for a selected product category at a selected run date.

Prompt details

You can set the following prompt filters in this report:

- Run Date
- Analysis Date

This prompt determines whether the Warranty start date is considered as the Date Sold, Date Manufactured, or Date Raw Materials Produced.

Drill behavior

There is a drill through link from the Product Code column to the QEWSL – Warranty Chart report.

Conditional formatting

Conditional formatting is applied to conditionally render the stoplight images for Replacement and for Wear out.

QEWSL - Warranty Chart

The quality early warning system life time (QEWSL) warranty chart reports the replacement rates for a specific product type and product code over a time period.

This chart is generated by IBM Predictive Maintenance and Quality daily. If the daily chart was not generated for the date that you select, the report is empty.

You can set the following prompt filters in this chart:

- Run Date
- Product Type
- Product Code

The chart heading contains the following information:

- Product code
- Last run date of the chart
- Period during which the product was shipped (start date and end date)
- Number of parts that were shipped over the period
- Number of parts that failed over the period
- Replacements per machine month of service over the period

Note: This chart is not an IBM Cognos Report Studio report so you cannot modify it in Report Studio.

Replacement rate chart

This chart has a dual x axis that shows vintage number and cumulative number of machine months of service. Vintage number is the day number that the part was shipped during the period. Cumulative number of machine months is the total number of machine months of service that is accrued by the population of

machines that have the parts installed. The y axis shows the replacement rate of the product per machine month. A data point on the chart indicates the replacement rate for a vintage. The Acceptable Level is a horizontal line on the chart that shows the acceptable replacement rate.

If the severity of wear out conditions is greater than zero, the chart contains a curve corresponding to monitoring wear out conditions. The levels of wear out index that is based on vintages summarized monthly corresponds to the y-axis on the chart.

Evidence chart

This chart monitors the reliability or characteristics of the lifetime of a part. The chart has a dual x axis that shows vintage number and cumulative number of machine months of service. Vintage number is the day number that the part was shipped as part of a machine. Cumulative machine months is the number of machine months of service. Cumulative machine month is shown on the x-axis. The y-axis shows the level of evidence that the underlying process replacement rate is unacceptable. It is computed by using a weighted cumulative sum (CUSUM) formula.

Threshold H is a horizontal line that shows the replacement rate threshold value. The CUSUM values that are higher than Threshold H are displayed as triangles on the chart. The triangles indicate unacceptable process levels in the data. The vertical dotted line indicates the last time that the vintage number had an unacceptable replacement rate per machine month.

If the severity of wear out conditions is greater than zero, the chart contains a curve corresponding to monitoring wear out conditions. The wear out curve is shown together with the corresponding threshold.

Summary list

The summary list header contains the same information as the chart heading. The summary list shows detailed information by vintage number. It includes date, number of parts that were tested, total quantity, and other data.

Quality dashboard - Parametric

The Quality dashboard - Parametric report provides an overview of the state of products at a selected run date for a variable.

Prompt details

You can set the following prompt filters in this report:

- Run Date
- Variable

Drill behavior

You can drill up and drill down on the Product column. There is a drill through link from the Product column to the Quality Dashboard – Parametric Detail History report.

Quality dashboard - Parametric Detail History

The Quality dashboard - Parametric Detail History report provides details about the state of products and the various threshold values for a selected product category, at a selected run date for a variable.

Prompt details

You can set the following prompt filters in this report:

- Run Date
- Variable

Drill behavior

There is a drill through link from the Product Code column to the QEWSV – Parametric Chart report.

Conditional formatting

Conditional formatting is applied to conditionally render the stoplight images for the Early Warning System (EWS) stoplights.

QEWSV - Parametric chart

The QEWSV - Parametric chart report is used for monitoring variable-type data and CUSUM values that are obtained from the QEWSV batch along with threshold levels.

The report is designed to support five different validation types: Material Validation, Process-Resource Validation, Production Batch Validation, Resource Health Check, and Location Suitability.

Prompt details

You can set the following prompt filters in this report:

- Run Date
- Validation Type
 - Production Batch is the supported default sub use case.
- Material
 - Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
 - When Validation Type is MVARIABLE, this prompt is shown. Otherwise, it is hidden.
- Process
 - Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
 - When Validation Type is PRVARIABLE, this prompt is shown. Otherwise, it is hidden.
- Resource
 - Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
 - When Validation Type is PRVARIABLE or RVARIABLE, this prompt is shown. Otherwise, it is hidden.

- Location
Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
When Validation Type is LVARIABLE, this prompt is shown. Otherwise, it is hidden.
- Product Type
Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
When Validation Type is PBVARIABLE, this prompt is shown. Otherwise, it is hidden.
- Product Code
Depending on the Validation type prompt, this prompt is shown or hidden. This prompt is cascaded from Validation Type.
When Validation Type is PBVARIABLE, this prompt is shown. Otherwise, it is hidden.
- Variable Type
This prompt represents the measurement type.

Drill behavior

The Material, Location, Process, Resource, Product Type, and Product Code prompts are conditionally displayed based on the Validation Type prompt selection.

TopN Failure Analysis Report

This report shows the profiles that contribute to the failure of a resource. Each profile has an importance value that is expressed as a percentage. The total of the importance values displayed on the report is 100%.

The profile is indicated by the x axis. The importance value is indicated on the y axis. Each profile is represented by a bar on the chart. The higher the importance value, the more that the profile contributes to the failure of the resource.

The curved line on the chart indicates the cumulative importance value.

You can set the following prompt filters in this chart:

- Location
- Resource Sub Type
- Resource
- Resource Code
- Event Code

You can also access this report from the Maintenance Health and Failure Detail Report. For more information, see “Maintenance Overview Dashboard” on page 160.

Drill through to the statistical process control reports

Select a profile from the **Analyze Profile Variable** list. Click a link to one of the statistical process control (SPC) reports.

Note: The raw measurement type for the profile is passed to the SPC report.

Appendix A. 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).

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

Report output

In IBM Cognos Administration, you can enable system-wide settings to create accessible report output. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*. In IBM Cognos Report Studio, you can enable settings to create accessible output for individual reports. For more information, see the *IBM Cognos Report Studio User Guide*. You can access the previously mentioned documents at IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

Appendix B. The flat file API

Use the flat file application programming interface (API) to supply and modify IBM Predictive Maintenance and Quality master data.

The IBM Predictive Maintenance and Quality API supports the **upsert** operation.

The **upsert** operation attempts to update an existing row. If the matching row cannot be found, a new row is created that uses the values in the input record.

All the values for the row must be included even if only a single value of the row is being changed.

An IS_ACTIVE indicator is used to mark records as no longer in use (IS_ACTIVE = 0).

The IS_ACTIVE indicator is not used to take any decisions while loading the master or event data. For example, while loading a resource, if the associated location has the following indicator: IS_ACTIVE=0, that resource is loaded and associated with that location. Similarly if the event is reported by the resource with IS_ACTIVE=0, the event is processed and stored in the datastore.

Master data in the API

Use master data to supply IBM Predictive Maintenance and Quality with information about the context in which events occur.

The following records are supported by the master data section of the application programming interface (API). They are listed in alphabetical order but functionally they fall into one of four logical groups:

- Resource-related records include the location, resource, and resource_type records
- Process-related records include the batch_batch, process, product, and production_batch records
- Material-related records include the material and material_type records
- Other records can be related to both devices and processes. These records include the group_dim, source_system, and supplier records

No Delete operation is supported for master data. The upsert API can be used to mark a master data row as no longer active. In this case, the item in the row is no longer used in reports.

Load order

Some tables include references to rows in other tables. A row must be loaded before it can be referenced from another table.

The language and tenant tables must be loaded before any other data is loaded. The language_cd and tenant_cd rows are referenced in many tables. The values that are supplied for the language_cd and tenant_cd rows must reference rows already present in the language and tenant tables.

In addition, the rows of some tables refer to other rows in the same table, for example, parent rows. The referenced rows must be added before the rows that reference them.

Master files must be loaded sequentially.

The following table lists tables that contain references to other tables.

Table 49. Tables that must exist before other tables can be loaded

Table	Prerequisite tables
batch_batch	production_batch
material	material_type, supplier
process	process (parent process) Note: No circular relationship is allowed. That is, a process_code cannot be a parent to itself.
production_batch	product
resource	group_dim, location, resource (parent resource)
profile_variable	measurement_type, material_type

batch_batch

Creates a many-to-many relationship between production batches.

Use the **batch_batch** for batch traceability so that batches that share materials can be enumerated when a defect is found at any point. Every batch must relate to every batch in its lineage for full traceability.

For example, batch 1 splits into 2 and 3, and batch 3 splits into 4 and 5.

batch_batch holds these pairs:

1,1 1,2 1,3 1,4 1,5 2,1 2,3 3,1 3,2 3,4 3,5 4,1 4,3 4,5 5,1 5,3 5,4

The fields in the **batch_batch** table are listed in the following table.

Table 50. Fields in the batch_batch table

Field	Type	Comments
production_batch_cd	string(50)	Required
related_production_batch_cd	string(50)	Required

batch_batch code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT PB1.PRODUCTION_BATCH_CD, PB2.PRODUCTION_BATCH_CD FROM
SYSREC.MASTER_BATCH_BATCH M JOIN SYSREC.MASTER_PRODUCTION_BATCH PB1 ON
M.PRODUCTION_BATCH_ID = PB1.PRODUCTION_BATCH_ID JOIN
SYSREC.MASTER_PRODUCTION_BATCH PB2 ON M.RELATED_PRODUCTION_BATCH_ID =
PB2.PRODUCTION_BATCH_ID;
```

event_code

Contains codes for alarms, failures, issues, and so on.

When an event arrives with a measurement type which has an event code indicator of 1, the text from the **event_observation_text** value is assumed to contain an event code. The measurement type of the event defines the **event_code_set** value.

The fields in the **event_code** table are listed in the following table.

Table 51. Fields in the **event_code** table

Field	Type	Comments
event_code_set	string(50)	Required
event_code_set_name	string(200)	Required
event_code	string(50)	Required
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

event_code code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.EVENT_CODE_SET, M.EVENT_CODE_SET_NAME, M.EVENT_CODE, L.LANGUAGE_CD,
T.TENANT_CD FROM SYSREC.MASTER_EVENT_CODE M JOIN SYSREC.LANGUAGE L ON
M.LANGUAGE_ID = L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID =
T.TENANT_ID;
```

group_dim

Provides classifications for resources.

Up to five classifications are possible for each resource. The classifications vary depending on how IBM Predictive Maintenance and Quality is used. For example, a classification may be manufacturer or organization.

The fields for the **group_dim** table are listed in the following table.

Table 52. Fields in the **group_dim** table

Field	Type	Comments
group_type_cd	string(50)	Required
group_type_name	string(200)	Required
group_member_cd	string(50)	Required
group_member_name	string(200)	Required

Table 52. Fields in the `group_dim` table (continued)

Field	Type	Comments
<code>language_cd</code>	<code>string(50)</code>	Optional. This value must reference a row in the language table.
<code>tenant_cd</code>	<code>string(50)</code>	Optional. This value must reference a row in the tenant table.

group_dim code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.GROUP_TYPE_CODE, M.GROUP_TYPE_TEXT, M.GROUP_MEMBER_CODE,
M.GROUP_MEMBER_TEXT, L.LANGUAGE_CD, T.TENANT_CD FROM SYSREC.MASTER_GROUP_DIM M
JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID = L.LANGUAGE_ID
JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

language

Contains the list of supported languages.

The fields in the **language** table are listed in the following table.

Table 53. Fields in the `language` table

Field	Type	Comments
<code>language_cd</code>	<code>string(50)</code>	Required. For example, EN
<code>language_name</code>	<code>string(200)</code>	Required. For example, English.
<code>DEFAULT_IND</code>	0 or 1	Optional. A value of 1 indicates that this language is the default language for the system. No value, or a value of 0, indicates the language is not the default.

language code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line.

```
SELECT LANGUAGE_CD, LANGUAGE_NAME, DEFAULT_IND FROM SYSREC.LANGUAGE;
```

New languages and tenants

After you have added new languages, or new tenants, you must populate the NA rows in the database for all new valid combinations of language and tenant. See the following example.

```
db2 "call SCHEMA.POP_NA( 'LANGUAGE_CD' , 'LANGUAGE_NAME' , 'TENANT_CD' , 'TENANT_NAME' )"
```

Where schema is a valid DB2 schema, such as db2inst1.

location

The location of a resource or event.

The location can be as specific, such as a room in a factory or general, such as a mine site.

The fields in the **location** table are listed in the following table.

Table 54. Fields in the location table

Field	Type	Comments
location_cd	string(50)	Required
location_name	string(200)	Required
region_cd	string(50)	Optional. The region_cd and region_name parameters must be supplied together.
region_name	string(200)	Optional
country_cd	string(50)	Optional. The country_cd and country_name parameters must be supplied together.
country_name	string(200)	Optional
state_province_cd	string(50)	Optional. The state_province_cd and state_province_name parameters must be supplied together.
state_province_name	string(200)	Optional
city_name	string(200)	Optional
latitude	decimal (in signed decimal degrees. N is + and S is -)	Optional
longitude	decimal (in signed decimal degrees. E is + and W is -)	Optional
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.
IS_ACTIVE	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

location code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.LOCATION_CD, M.LOCATION_NAME, M.REGION_CD, M.REGION_NAME, M.COUNTRY_CD,
M.COUNTRY_NAME, M.STATE_PROVINCE_CD, M.STATE_PROVINCE_NAME, M.CITY_NAME,
M.LATITUDE, M.LONGITUDE, L.LANGUAGE_CD, T.TENANT_CD, M.ISACTIVE FROM
SYSREC.MASTER_LOCATION M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

material

Defines the material that is used for an event.

The fields in the **material** table are defined as a specific instance of a material type, including a link to the supplier. It can be material that is used in a repair or material that is used in a production process.

The fields in the **material** table are listed in the following table.

Table 55. Fields in the **material** table

Field	Type	Comments
material_cd	string(50)	Required
material_name	string(200)	Required
material_type_cd	string(50)	Required
supplier_cd	string(50)	Required
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.
IS_ACTIVE	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

material code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.MATERIAL_CD, M.MATERIAL_NAME, MT.MATERIAL_TYPE_CD, S.SUPPLIER_CD,
L.LANGUAGE_CD, T.TENANT_CD, M.ISACTIVE FROM SYSREC.MASTER_MATERIAL M
JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID = L.LANGUAGE_ID JOIN
SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID JOIN
SYSREC.MASTER_MATERIAL_TYPE MT ON M.MATERIAL_TYPE_ID = MT.MATERIAL_TYPE_ID AND
M.LANGUAGE_ID = MT.LANGUAGE_ID JOIN SYSREC.MASTER_SUPPLIER S ON M.SUPPLIER_ID =
S.SUPPLIER_ID AND M.LANGUAGE_ID = S.LANGUAGE_ID;
```

material_type

A categorization of material by type.

Material type is material that is used in a repair, such as engine filters or parts, or it can be material that is used in a production process.

The fields in the **material_type** table are listed in the following table.

Table 56. Fields in the material_type table

Field	Type	Comments
material_type_cd	string(50)	Required
material_type_name	string(200)	Required
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

material_type code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.MATERIAL_TYPE_CD, M.MATERIAL_TYPE_NAME, L.LANGUAGE_CD, T.TENANT_CD FROM
SYSREC.MASTER_MATERIAL_TYPE M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

process

Represents a production process.

A process can be part of a hierarchy of processes.

The fields in the **process** table are listed in the following table.

Table 57. Fields in the process table

Field	Type	Comments
process_cd	string(50)	Required
process_name	string(200)	Required
parent_process_cd	string(50)	Optional
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

process code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.PROCESS_CD, M.PROCESS_NAME, P.PROCESS_CD AS PARENT_PROCESS_CD,
L.LANGUAGE_CD, T.TENANT_CD FROM SYSREC.MASTER_PROCESS M JOIN
SYSREC.LANGUAGE L ON M.LANGUAGE_ID = L.LANGUAGE_ID JOIN
SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID JOIN SYSREC.MASTER_PROCESS
P ON M.PARENT_PROCESS_ID = P.PARENT_PROCESS_ID AND M.LANGUAGE_ID = P.LANGUAGE_ID;
```

product

Defines the product being produced by the events.

The fields in the **product** table are listed in the following table.

Table 58. Fields in the product table

Field	Type	Comments
product_cd	string(50)	Required
product_name	string(200)	Required
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.
IS_ACTIVE	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

product code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.PRODUCT_CD, M.PRODUCT_NAME, L.LANGUAGE_CD, T.TENANT_CD, M.ISACTIVE FROM
SYSREC.MASTER_PRODUCT M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

production_batch

Contains information about product groupings during the production event.

A batch can split and merge throughout the production process, and so one batch can be related to many other batches.

The fields in the **production_batch** table are listed in the following table.

Table 59. Fields in the production_batch table

Field	Type	Comments
production_batch_cd	string(50)	Required
production_batch_name	string(200)	Required
product_cd	string(50)	Required
product_type_cd	string(50)	Required

Table 59. Fields in the `production_batch` table (continued)

Field	Type	Comments
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

production_batch code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.PRODUCTION_BATCH_CD, M.PRODUCTION_BATCH_NAME, P.PRODUCT_CD,
L.LANGUAGE_CD, T.TENANT_CD FROM SYSREC.MASTER_PRODUCTION_BATCH M JOIN
SYSREC.LANGUAGE L ON M.LANGUAGE_ID = L.LANGUAGE_ID JOIN
SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID JOIN SYSREC.MASTER_PRODUCT
P ON M.PRODUCT_ID = P.PRODUCT_ID AND M.LANGUAGE_ID = P.LANGUAGE_ID;
```

profile_calculation

These records define a set of profile calculation names.

Profile calculations aggregate event values into KPIs and Profiles.

The fields in the **profile_calculation** table are listed in the following table.

Table 60. Fields in the `profile_calculation` table

Field	Type	Comments
profile_calculation_name	string(200)	Required
language_cd	string(50)	Optional
tenant_cd	string(50)	Optional

profile_calculation code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.PROFILE_CALCULATION_NAME, T.TENANT_CD FROM
SYSREC.MASTER_PROFILE_CALCULATION M JOIN SYSREC.TENANT T ON M.TENANT_ID
= T.TENANT_ID;
```

resource

Defines resources of type asset or agent. Asset or agent are the only resource types allowed.

An asset is a piece of equipment. An agent is the operator of the equipment. Some asset resources can form a hierarchy. For example, a truck is a parent of a tire.

Parent resources must be loaded before child resources. Resources cannot be their own parent.

More specific types of resources can be named in the resource_sub_type column.

The fields in the **resource** table are listed in the following table.

Table 61. Fields in the resource table

Field	Type	Comments
resource_cd1	string(50)	Optional. Used to provide serial Number. Either resource_cd1 and resource_cd2 are required or operator_cd is required.
resource_cd2	string(50)	Optional. Used to provide model information of resource.
operator_cd	string(50)	Optional
resource_name	string(500)	Required
resource_type_cd	string(50)	Required
resource_sub_type	string(50)	Optional
parent_resource_serial_no	string(50)	Optional. The parent_resource_serial_no and parent_resource_model parameters must be supplied together.
parent_resource_model	string(50)	Optional
parent_resource_operator_cd	string(50)	Optional
standard_production_rate	decimal	Optional
production_rate_uom	string(40)	Optional
preventative_maintenance_interval	decimal	Optional
group_dim_type_cd_1	string(50)	Optional. The type and a member must be supplied together.
group_dim_member_cd_1	string(50)	Optional
group_dim_type_cd_2	string(50)	Optional
group_dim_member_cd_2	string(50)	Optional
group_dim_type_cd_3	string(50)	Optional
group_dim_member_cd_3	string(50)	Optional
group_dim_type_cd_4	string(50)	Optional
group_dim_member_cd_4	string(50)	Optional
group_dim_type_cd_5	string(50)	Optional
group_dim_member_cd_5	string(50)	Optional
location_cd	string(50)	Optional

Table 61. Fields in the resource table (continued)

Field	Type	Comments
language_cd	string(50)	Optional. This value must reference a row in the language table.
mfg_date	date	Optional. Manufacturing date of resource.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.
IS_ACTIVE	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

resource code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.RESOURCE_CD1, M.RESOURCE_CD2, M.RESOURCE_NAME,
       RT.RESOURCE_TYPE_CD,M.RESOURCE_SUB_TYPE, P.RESOURCE_CD1 AS
       PARENT_RESOURCE_CD1,P.RESOURCE_CD1 AS PARENT_RESOURCE_CD2,
       M.STANDARD_PRODUCTION_RATE, M.PRODUCTION_RATE_UOM,M.PREVENTIVE_MAINTENANCE_INTERVAL,
       G1.GROUP_TYPE_CD AS GROUP_TYPE_CD_1,G1.GROUP_MEMBER_CD AS GROUP_MEMBER_CD_1,
       G2.GROUP_TYPE_CD AS GROUP_TYPE_CD_2,G2.GROUP_MEMBER_CD AS GROUP_MEMBER_CD_2,
       G3.GROUP_TYPE_CD AS GROUP_TYPE_CD_3,G3.GROUP_MEMBER_CD AS GROUP_MEMBER_CD_3,
       G4.GROUP_TYPE_CD AS GROUP_TYPE_CD_4,G4.GROUP_MEMBER_CD AS GROUP_MEMBER_CD_4,
       G5.GROUP_TYPE_CD AS GROUP_TYPE_CD_5,G5.GROUP_MEMBER_CD AS GROUP_MEMBER_CD_5,
       LC.LOCATION_CD,M.MFG_DATE, L.LANGUAGE_CD,T.TENANT_CD, M.IS_ACTIVE FROM
       SYSREC.MASTER_RESOURCE M JOIN SYSREC.LANGUAGEL ON M.LANGUAGE_ID = L.LANGUAGE_ID JOIN
       SYSREC.TENANT T ON M.TENANT_ID =T.TENANT_ID LEFT OUTER JOIN SYSREC.MASTER_RESOURCE P ON
       M.PARENT_RESOURCE_ID =P.MASTER_RESOURCE_ID AND M.LANGUAGE_ID = P.LANGUAGE_ID JOIN
       SYSREC.MASTER_GROUP_DIM G1 ONM.GROUP_DIM_ID_1 = G1.MASTER_GROUP_DIM_ID AND
       M.LANGUAGE_ID = G1.LANGUAGE_ID JOINSYSREC.MASTER_GROUP_DIM G2 ON M.GROUP_DIM_ID_2 =
       G2.MASTER_GROUP_DIM_ID AND M.LANGUAGE_ID= G2.LANGUAGE_ID JOIN SYSREC.MASTER_GROUP_DIM G3
       ON M.GROUP_DIM_ID_3 =G3.MASTER_GROUP_DIM_ID AND M.LANGUAGE_ID = G3.LANGUAGE_ID JOIN
       SYSREC.MASTER_GROUP_DIM G4ON M.GROUP_DIM_ID_4 = G4.MASTER_GROUP_DIM_ID AND
       M.LANGUAGE_ID = G4.LANGUAGE_ID JOINSYSREC.MASTER_GROUP_DIM G5 ON M.GROUP_DIM_ID_5 =
       G5.MASTER_GROUP_DIM_ID AND M.LANGUAGE_ID= G5.LANGUAGE_ID JOIN SYSREC.MASTER_LOCATION LC
       ON M.LOCATION_ID = LC.MASTER_LOCATION_ID AND M.LANGUAGE_ID = LC.LANGUAGE_ID JOIN
       SYSREC.MASTER_RESOURCE_TYPE RT ONM.RESOURCE_TYPE_ID = RT.MASTER_RESOURCE_TYPE_ID AND
       M.LANGUAGE_ID = RT.LANGUAGE_ID;
```

resource_type

These records categorize resources.

The two supported resource types are asset and agent. An asset is a piece of equipment that is used in the production process. An agent is the operator of the equipment.

The fields in the **resource_type** table are listed in the following table.

Table 62. Fields in the `resource_type` table

Field	Type	Comments
<code>resource_type_cd</code>	string(50)	Required
<code>resource_type_name</code>	string(200)	Required
<code>language_cd</code>	string(50)	Optional. This value must reference a row in the language table.
<code>tenant_cd</code>	string(50)	Optional. This value must reference a row in the tenant table.

resource_type code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.RESOURCE_TYPE_CD, M.RESOURCE_TYPE_NAME, L.LANGUAGE_CD, T.TENANT_CD FROM
SYSREC.MASTER_RESOURCE_TYPE M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

source_system

Contains information about the system generating an event.

The fields in the **source_system** table are listed in the following table.

Table 63. Fields in the `source_system` table

Field	Type	Comments
<code>source_system_cd</code>	string(50)	Required.
<code>source_system_name</code>	string(200)	Required.
<code>language_cd</code>	string(50)	Optional. This value must reference a row in the language table.
<code>tenant_cd</code>	string(50)	Optional. This value must reference a row in the tenant table.
<code>IS_ACTIVE</code>	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

source_system code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.SOURCE_SYSTEM_CD, M.SOURCE_SYSTEM_NAME, L.LANGUAGE_CD, T.TENANT_CD,
M.ISACTIVE FROM SYSREC.MASTER_SOURCE_SYSTEM M JOIN SYSREC.LANGUAGE L ON
M.LANGUAGE_ID = L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID =
T.TENANT_ID;
```

supplier

Contains material supplier information.

The fields in the **supplier** table are listed in the following table.

Table 64. Fields in the **supplier** table

Field	Type	Comments
supplier_cd	string(50)	Required.
supplier_name	string(200)	Required.
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.
IS_ACTIVE	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

supplier code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.SUPPLIER_CD, M.SUPPLIER_NAME, L.LANGUAGE_CD, T.TENANT_CD, M.ISACTIVE
FROM SYSREC.MASTER_SUPPLIER M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

tenant

Contain the list of supported tenants.

The fields in the **tenant** table are listed in the following table.

Table 65. Fields in the **tenant** table

Field	Type	Comments
tenant_cd	string(50)	Required.
tenant_name	string(200)	Required.
DEFAULT_IND	0 or 1	Optional. A value of 0 indicates that the record is inactive. No value, or a value of 1, indicates that the record is active.

tenant code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line.

```
SELECT TENANT_CD, TENANT_NAME, DEFAULT_IND FROM SYSREC.TENANT;
```

For information on adding new languages and tenants, see the following information: “New languages and tenants” on page 178.

Changing the tenant code and name

You can rename the tenant code and name. For example, in the sample data, the tenant code and name by default is PMQ.

Procedure

1. Type the following command to connect to the **IBMPMQ** database by connecting to the DB2 node:

```
db2 "connect to IBMPMQ user user_name using password"
```

2. Type the following command:

```
db2 "update sysrec.master_tenant set tenant_code='CODE',
tenant_name='NAME' where tenant_code='PMQ'"
```

Where *CODE* is the tenant code, and *NAME* is the tenant name.

For example, the following code renames the tenant code to XY, and the tenant name to XY Ltd.

```
db2 "update sysrec.master_tenant set tenant_code='XY',
tenant_name='XY Ltd' where tenant_code='PMQ'"
```

3. Type the following command to commit the transaction:

```
db2 "commit"
```

4. Type the following command to disconnect from the database:

```
db2 "connect reset"
```

value_type

Defines the set of possible numerical observations, including actual, planned or forecast.

The fields for the **value_type** table are listed in the following table.

Table 66. Fields for the **value_type** table

Field	Type	Comments
value_type_cd	string(50)	Required
value_type_name	string(200)	Required
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

value_type code snippet

You can use the following SQL Code snippet to retrieve master data in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load master data, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.VALUE_TYPE_CD, M.VALUE_TYPE_NAME, L.LANGUAGE_CD, T.TENANT_CD FROM
SYSREC.MASTER_VALUE_TYPE M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.MASTER_TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

Metadata in the API

The following records are supported by the metadata section of the application programming interface (API). The records are listed in alphabetical order.

event_type

These records define a categorization of events.

Some examples of event types are measurement, alarm, and inspection.

The fields in the **event_type** table are listed in the following table.

Table 67. Fields in the event_type table

Field	Type	Comments
event_type_cd	string(50)	Required.
event_type_name	string(200)	Required.
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

event_type code snippet

You can use the following SQL Code snippet to retrieve metadata in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load metadata, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.EVENT_TYPE_CD, M.EVENT_TYPE_NAME, L.LANGUAGE_CD, T.TENANT_CD FROM
SYSREC.MASTER_EVENT_TYPE M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID
```

measurement_type

Contains all the measures and event code sets that can be observed for the **resource**, **process**, and **material** records.

Some examples of measurement types are engine oil pressure, ambient temperature, fuel consumption, conveyor belt speed, capping pressure, and so on.

In the case of measurement types where the **event_code_indicator** value is 1, there is a special class to capture failure codes, issue codes, and alarm codes as

event_code records. The **measurement_type_code** and **measurement_type_name** records become the **event_code_set** and **event_code_set_name** records respectively. This is a trigger to the event integration process to begin recording event codes from the **observation_text** record.

The fields for the **measurement_type** table are listed in the following table.

Table 68. Fields for the measurement_type

Field	Type	Comments
measurement_type_cd	string(50)	Required
measurement_type_name	string(200)	Required
unit_of_measure	string(100)	Optional
carry_forward_indicator	0 or 1	Optional
aggregation_type	string(100)	Optional
event_code_indicator	0 or 1	Optional
language_cd	string(50)	Optional. This value must reference a row in the language table.
tenant_cd	string(50)	Optional. This value must reference a row in the tenant table.

measurement_type code snippet

You can use the following SQL Code snippet to retrieve metadata in the format that is required by the **upsert** API.

For example, if you lost the original files that are used to load metadata, you can use the snippet to retrieve the data, make changes, and submit the changes by using the **upsert** API.

The command must be on a single line, not as shown here.

```
SELECT M.MEASUREMENT_TYPE_CD, M.MEASUREMENT_TYPE_NAME, M.UNIT_OF_MEASURE,
M.CARRY_FORWARD_INDICATOR, M.AGGREGATION_TYPE, M.EVENT_CODE_INDICATOR,
L.LANGUAGE_CD, T.TENANT_CD FROM SYSREC.MASTER_MEASUREMENT_TYPE M JOIN
SYSREC.LANGUAGE L ON M.LANGUAGE_ID = L.LANGUAGE_ID JOIN
SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID;
```

profile_variable

These records relate the **measurement_type**, **resource_type**, and **material_type** values to profile calculations.

The fields in the **profile_variable** table are listed in the following table.

Table 69. Fields in the profile_variable table

Field	Type	Comments
profile_variable_cd	string(50)	Required
profile_variable_name	string(200)	Required
profile_calculation_name	string(200)	Required
measurement_type_cd	string(50)	Required
resource_type_cd	string(50)	Optional
material_type_cd	string(50)	Optional

Table 69. Fields in the `profile_variable` table (continued)

Field	Type	Comments
<code>profile_units</code>	string(100)	Optional
<code>comparison_string</code>	string(200)	Optional
<code>low_value_date</code>	datetime	Optional
<code>high_value_date</code>	datetime	Optional
<code>low_value_number</code>	decimal	Optional
<code>high_value_number</code>	decimal	Optional
<code>kpi_indicator</code>	0 or 1	Optional. To disable a profile variable, set its <code>kpi_indicator</code> and <code>profile_indicator</code> to 0
<code>profile_indicator</code>	0 or 1	Optional. To disable a profile variable, set its <code>kpi_indicator</code> and <code>profile_indicator</code> to 0
<code>data_type</code>	string(100)	Optional
<code>aggregation_type</code>	string(100)	Optional
<code>carry_forward_indicator</code>	0 or 1	Optional
<code>process_indicator</code>	0 or 1	Optional
<code>variance_multiplier</code>	-1 or 1	Required. A value of 1 indicates that a higher measurement value is preferred. A value of -1 that a lower value is preferred.
<code>tenant_cd</code>	string(50)	Optional. This value must reference a row in the tenant table.

Due to references from the KPI and Profile tables, the `upsert` API for a `profile_variable` only allows the values of the following fields to be updated

- `profile_units`
- `comparison_string`
- `low_value_date`
- `high_value_date`
- `low_value_number`
- `kpi_indicator`
- `profile_indicator`
- `data_type`
- `aggregation_type`
- `process_indicator`
- `profile_variable_name`

profile_variable code snippet

You can use the following SQL Code snippet to retrieve metadata in the format that is required by the `upsert` API.

For example, if you lost the original files that are used to load metadata, you can use the snippet to retrieve the data, make changes, and submit the changes by using the `upsert` API.

The command must be on a single line, not as shown here.

```

SELECT M.PROFILE_VARIABLE_CD, M.PROFILE_VARIABLE_NAME, PC.PROFILE_CALCULATION_NAME,
MSRT.MEASUREMENT_TYPE_CD, RT.RESOURCE_TYPE_CD, MT.MATERIAL_TYPE_CD, M.PROFILE_UNITS,
M.COMPARISON_STRING, M.LOW_VALUE_DATE, M.HIGH_VALUE_DATE, M.LOW_VALUE_NUMBER,
M.HIGH_VALUE_NUMBER, M.KPI_INDICATOR, M.PROFILE_INDICATOR, M.DATA_TYPE,
M.AGGREGATION_TYPE, M.CARRY_FORWARD_INDICATOR, M.PROCESS_INDICATOR,
M.VARIANCE_MULTIPLIER, L.LANGUAGE_CD, T.TENANT_CD FROM
SYSREC.MASTER_PROFILE_VARIABLE M JOIN SYSREC.LANGUAGE L ON M.LANGUAGE_ID =
L.LANGUAGE_ID JOIN SYSREC.TENANT T ON M.TENANT_ID = T.TENANT_ID JOIN
SYSREC.MASTER_PROFILE_CALCULATION PC ON M.PROFILE_CALCULATION_ID =
PC.PROFILE_CALCULATION_ID JOIN SYSREC.MASTER_MEASUREMENT_TYPE MSRT ON
M.MEASUREMENT_TYPE_ID = MSRT.MEASUREMENT_TYPE_ID AND M.LANGUAGE_ID =
MSRT.LANGUAGE_ID JOIN SYSREC.MASTER_RESOURCE_TYPE RT ON M.RESOURCE_TYPE_ID =
RT.RESOURCE_TYPE_ID AND M.LANGUAGE_ID = RT.LANGUAGE_ID JOIN
SYSREC.MASTER_MATERIAL_TYPE MT ON M.MATERIAL_TYPE_ID = MT.MATERIAL_TYPE_ID AND
M.LANGUAGE_ID = MT.LANGUAGE_ID;

```

Mandatory profile variables and measurement types

To be able to process some events, you must load mandatory profile variables and measurement types.

Mandatory profile variables

The following profile variables must be loaded:

HS Required for health score related calculations.

RC Required for calculations that are related to the recommendations count.

You can see examples in the `profile_variable_upsert_sample_pmq.csv` file. This is installed on the Enterprise Service Bus (ESB) node computer in the `/var/PMQ/MQSIFileInput/PMQSampleData/Sample_PMQ/MasterData-Set2` folder.

Define profile variables that are based on the design of the IBM Cognos Business Intelligence reports and predictive models.

For example, for the sample models shipped with IBM Predictive Maintenance and Quality, the following profile variables and corresponding measurement types must be defined for the field `profile_variable_cd`:

- AC
- ATIME
- CELLLDX
- CELLLDXX
- CLTX
- CLTXX
- FAIL
- HS
- INSP
- ITIME
- OPHD
- QTY
- RC
- REPC
- REPT
- SETX
- SETXX

- SLTX
- SLTXX

Mandatory measurement types

The following measurement types must be loaded:

HS Required for health score related calculations.

You can see examples of these measurement types in the `measurement_type_upsert_sample_pmq.csv` file. This is installed on the Enterprise Service Bus (ESB) node computer in the `/var/PMQ/MQSIFileInput/PMQSampleData/Sample_PMQ/MasterData-Set1`.

Sample health score and IBM Analytical Decision Management services are configured for these measurement types:

- FAIL
- INSP
- LUBE
- OPHR
- PRS1
- PRS2
- PRS3
- RELH
- REPT
- REPX
- RPM
- R_B1
- R_F1
- TEMP

For health score, define profile variables with the profile calculations for the listed measurement types:

- Measurement of Type
- Measurement Above Limit (except for FAIL)
- Measurement Below Limit (except for FAIL)

Remove master data

Normally master data is not deleted from the analytic database. During testing and development, master data that is not referenced can be removed.

Sample code to remove master data

The following SQL code is an example and must be modified.

```
-- batch batch
DELETE FROM SYSREC.MASTER_BATCH_BATCH M WHERE
M.PRODUCTION_BATCH_ID = (SELECT PB1.PRODUCTION_BATCH_ID FROM
SYSREC.MASTER_PRODUCTION_BATCH PB1
JOIN SYSREC.LANGUAGE L ON PB1.LANGUAGE_ID = L.LANGUAGE_ID
JOIN SYSREC.TENANT T ON PB1.TENANT_ID = T.TENANT_ID WHERE
PB1.PRODUCTION_BATCH_CD = '1007' AND L.LANGUAGE_CD = 'EN' AND T.TENANT_CD = 'PMQ')
AND
M.RELATED_PRODUCTION_BATCH_ID = (SELECT PB2.PRODUCTION_BATCH_ID FROM
```

```

SYSREC.MASTER_PRODUCTION_BATCH PB2
JOIN SYSREC.LANGUAGE L ON PB2.LANGUAGE_ID = L.LANGUAGE_ID
JOIN SYSREC.TENANT T ON PB2.TENANT_ID = T.TENANT_ID WHERE
PB2.PRODUCTION_BATCH_CD = '1010' AND L.LANGUAGE_CD = 'EN' AND T.TENANT_CD = 'PMQ');

-- event code
DELETE FROM SYSREC.MASTER_EVENT_CODE M WHERE
M.EVENT_CODE_SET = 'FAIL' AND
M.EVENT_CODE = 'X101' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- event type
DELETE FROM SYSREC.MASTER_EVENT_TYPE M WHERE
M.EVENT_TYPE_CD = 'ALARM' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- group dim
DELETE FROM SYSREC.MASTER_GROUP_DIM M WHERE
M.GROUP_TYPE_CODE = 'ORG' AND
M.GROUP_MEMBER_CODE = 'C1' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- location
DELETE FROM SYSREC.MASTER_LOCATION M WHERE
M.LOCATION_CD = 'Room1' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- material
DELETE FROM SYSREC.MASTER_MATERIAL M WHERE
M.MATERIAL_CD = '20390' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- material type
DELETE FROM SYSREC.MASTER_MATERIAL_TYPE M WHERE
M.MATERIAL_TYPE_CD = 'PROD' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- measurement type
DELETE FROM SYSREC.MASTER_MEASUREMENT_TYPE M WHERE
M.MEASUREMENT_TYPE_CD = 'SET' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
T.TENANT_CD = 'PMQ');

-- process hierarchy
DELETE FROM SYSREC.PROCESS_HIERARCHY M WHERE
M.PROCESS_ID = (SELECT P.PROCESS_ID FROM SYSREC.MASTER_PROCESS P WHERE
P.PROCESS_CD = 'SET') AND

```

```

M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- process
DELETE FROM SYSREC.MASTER_PROCESS M WHERE
M.PROCESS_CD = 'SET' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- product
DELETE FROM SYSREC.MASTER_PRODUCT M WHERE
M.PRODUCT_CD = '2190890' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- production_batch
DELETE FROM SYSREC.MASTER_PRODUCTION_BATCH M WHERE
M.PRODUCTION_BATCH_CD = '1000' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- profile variable
DELETE FROM SYSREC.MASTER_PROFILE_VARIABLE M WHERE
M.PROFILE_VARIABLE_CD = 'SET' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- resource hierarchy
DELETE FROM SYSREC.RESOURCE_HIERARCHY M WHERE
M.RESOURCE_ID = (SELECT R.RESOURCE_ID FROM SYSREC.MASTER_RESOURCE R WHERE
  R.SERIAL_NO = '13580' AND R.MODEL = 'M100' ) AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- resource
DELETE FROM SYSREC.MASTER_RESOURCE M WHERE
M.SERIAL_NO = '13580' AND
M.MODEL = 'M100' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- source system
DELETE FROM SYSREC.MASTER_SOURCE_SYSTEM M WHERE
M.SOURCE_SYSTEM_CD = 'PREDMAIT' AND
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE
  L.LANGUAGE_CD = 'EN') AND
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE
  T.TENANT_CD = 'PMQ');

-- supplier
DELETE FROM SYSREC.MASTER_SUPPLIER M WHERE
M.SUPPLIER_CD = 'WS' AND

```

```
M.LANGUAGE_ID = (SELECT L.LANGUAGE_ID FROM SYSREC.LANGUAGE L WHERE  
L.LANGUAGE_CD = 'EN') AND  
M.TENANT_ID = (SELECT T.TENANT_ID FROM SYSREC.TENANT T WHERE  
T.TENANT_CD = 'PMQ');
```

Note:

The contents of the SYSREC.LANGUAGE, SYSREC.MASTER_PROFILE_CALCULATION, SYSREC.TENANT, SYSREC.MASTER_VALUE_TYPE, and SYSREC.MASTER_RESOURCE_TYPE tables are normally not deleted when master data is removed.

Appendix C. IBM Cognos Framework Manager model description

IBM Predictive Maintenance and Quality uses IBM Cognos Framework Manager to model the metadata for reports.

IBM Cognos Framework Manager is a metadata modeling tool that drives query generation for IBM Cognos software. A model is a collection of metadata that includes physical information and business information for one or more data sources. IBM Cognos software enables performance management on normalized and denormalized relational data sources and a variety of OLAP data sources.

For information on modifying or creating Framework Manager models, see the *IBM Cognos Framework Manager User Guide* and *IBM Cognos Framework Manager - Guidelines for Modeling Metadata*. These documents are available at IBM Cognos Business Intelligence Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSEP7J>).

The Framework Manager model consists of three layers:

- Database layer
- Logical layer
- Dimensional layer

Each of these layers is in a separate namespace. The dimensional layer is published to a package for use in reporting.

IBM Cognos Framework Manager model database layer

The physical, or database, layer contains a database query subject for every table in the physical data model. The database layer also contains alias shortcuts, which behave as if they were a copy of the original object with completely independent behavior.

The alias shortcuts are provided for two situations:

- To eliminate ambiguity for an entity that may be involved in multiple relationships, including the following items:
 - location and location (resource)
 - material_type and material_type (profile_variable)
 - resource_type and resource_type (profile_variable)
 - production_batch and production_batch (related)
- To enable you to query multiple copies of the same table in different roles, including the group_dim_1 to 5 values

If a database entity includes the language_id or tenant_id attributes, the database query subject includes a parameterized filter for each that selects only one tenant or language. Language is based on the used locale settings. Localization is implemented for the FM model as well. Users can select the language of their choice from the Active Language drop down menu and change the model language.

The database layer contains all of the entity relationships. The central entities are largely modeled in star or snowflake schemas, shown in the following diagrams. These parameters must be set after master data has been loaded or reloaded and before publishing the package. If these parameters are not set correctly, no data will be returned in the reports. To change the values, simply open the parameter map, double click on the value for each parameter and type over it.

A parameter map for language supports the localization of report data. Language codes for English (EN), Simplified Chinese (SC), Traditional Chinese (TC), French (FR), Japanese (JP), and Portuguese (Brazil)(PT) are configured in the parameter map.

Typically the central fact has cardinality 1,N and related objects are 1,1, in order to eliminate the need for relationships outside of the database layer. All joins are modeled as inner joins on the understanding that the data integration layer populates a default value for all references in the absence of a valid value.

The following diagram shows the star schema for the event_observation table.

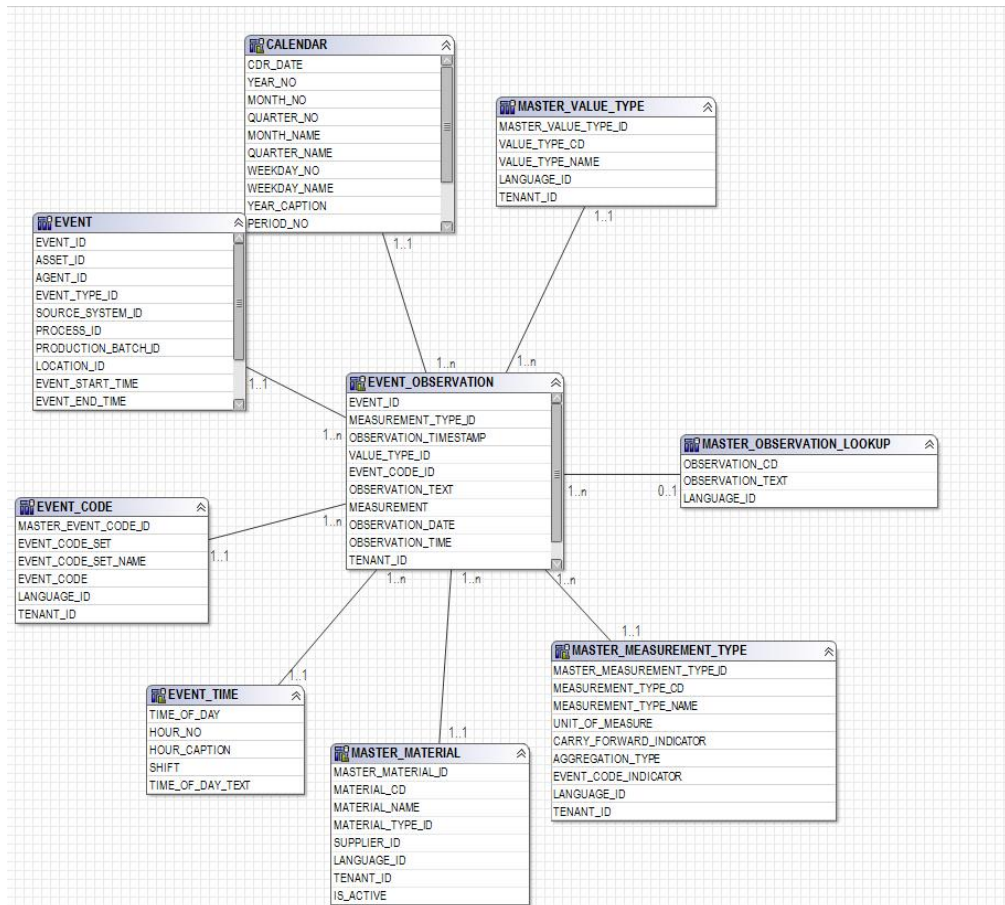


Figure 61. The event_observation star schema

The following diagram shows the star schema for the resource_profile table.

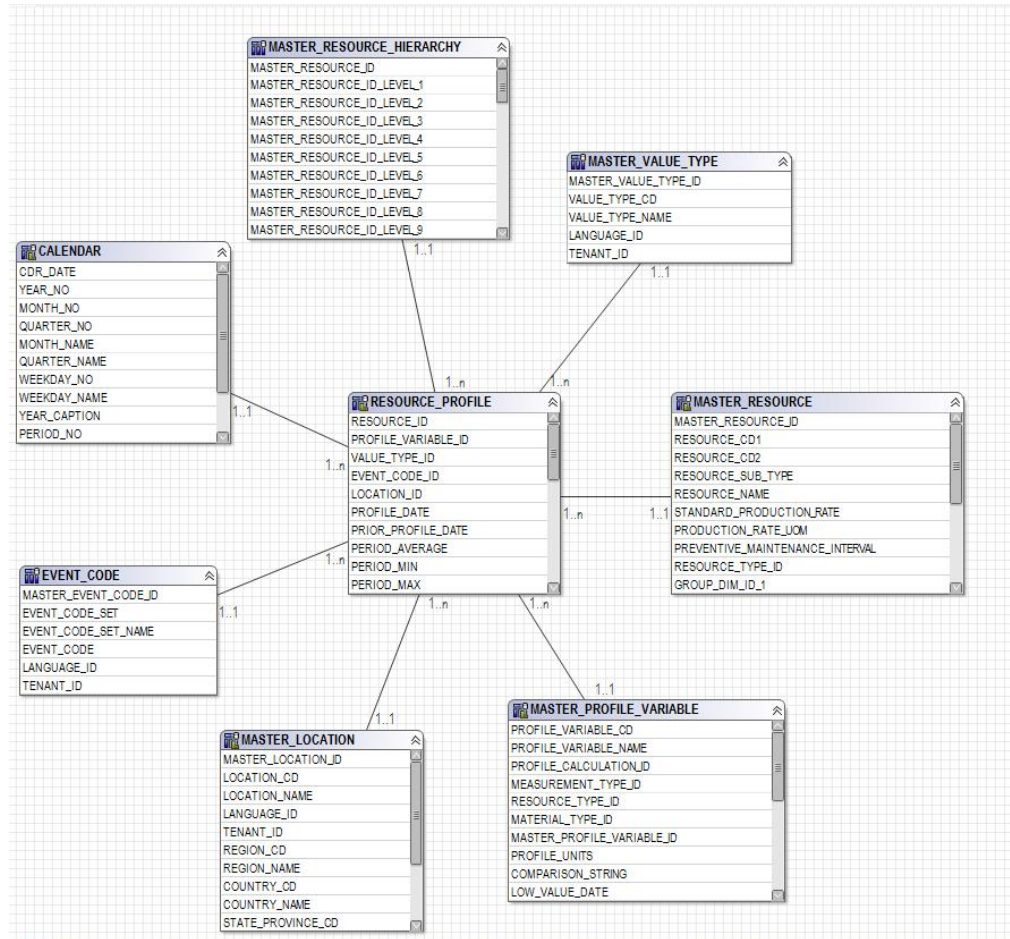


Figure 62. The resource_profile star schema

The following diagram shows the star schema for the resource_kpi table.

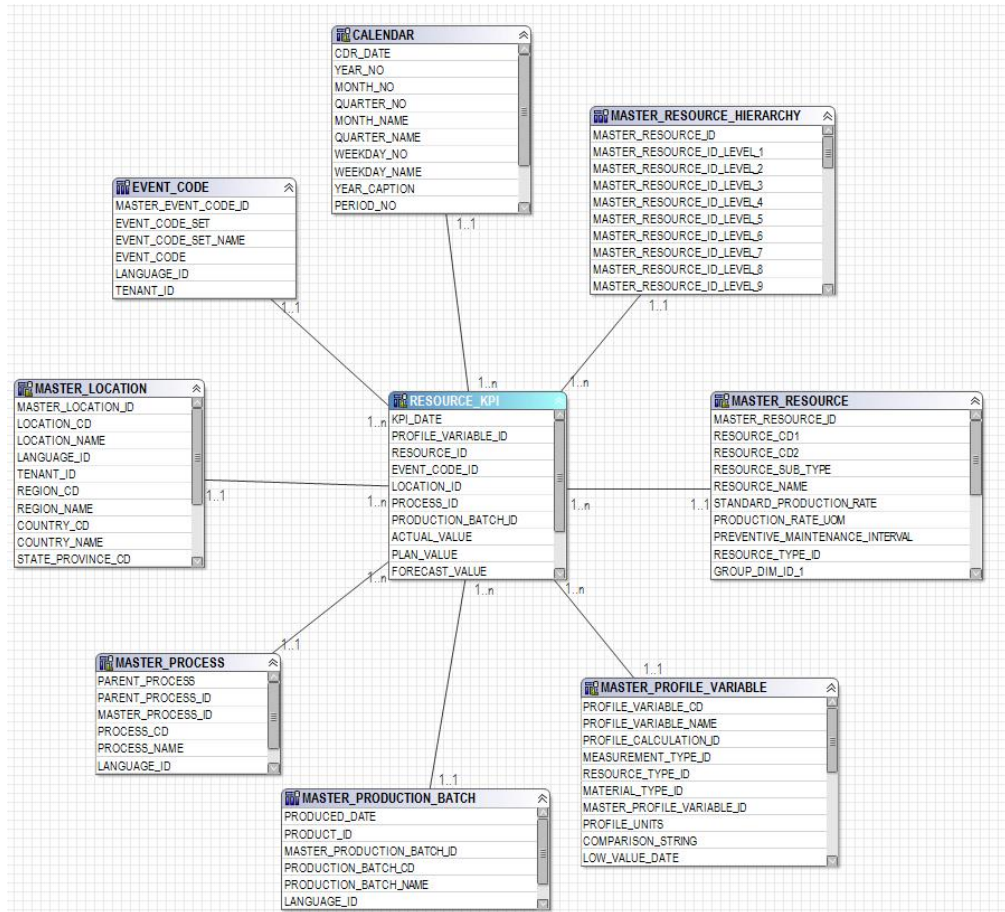


Figure 63. The resource_kpi star schema

The following diagram shows the star schema for the material_profile table.

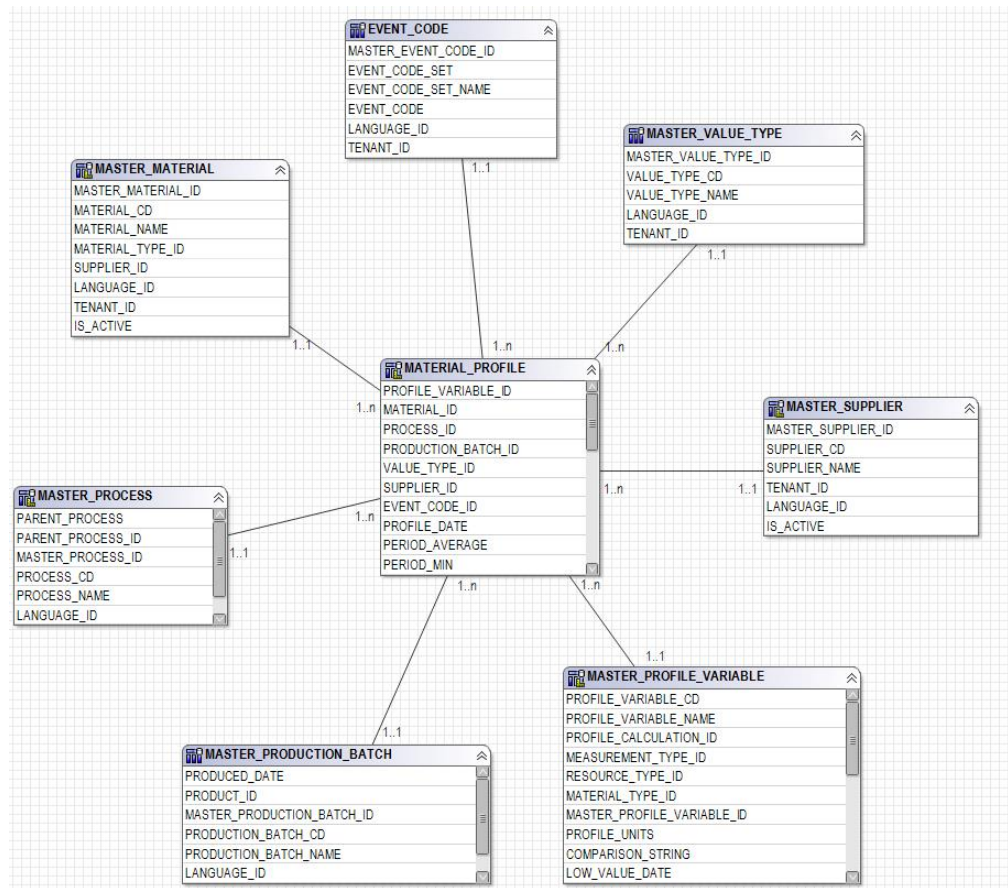


Figure 64. The material_profile star schema

The following diagram shows the star schema for the process_profile table.

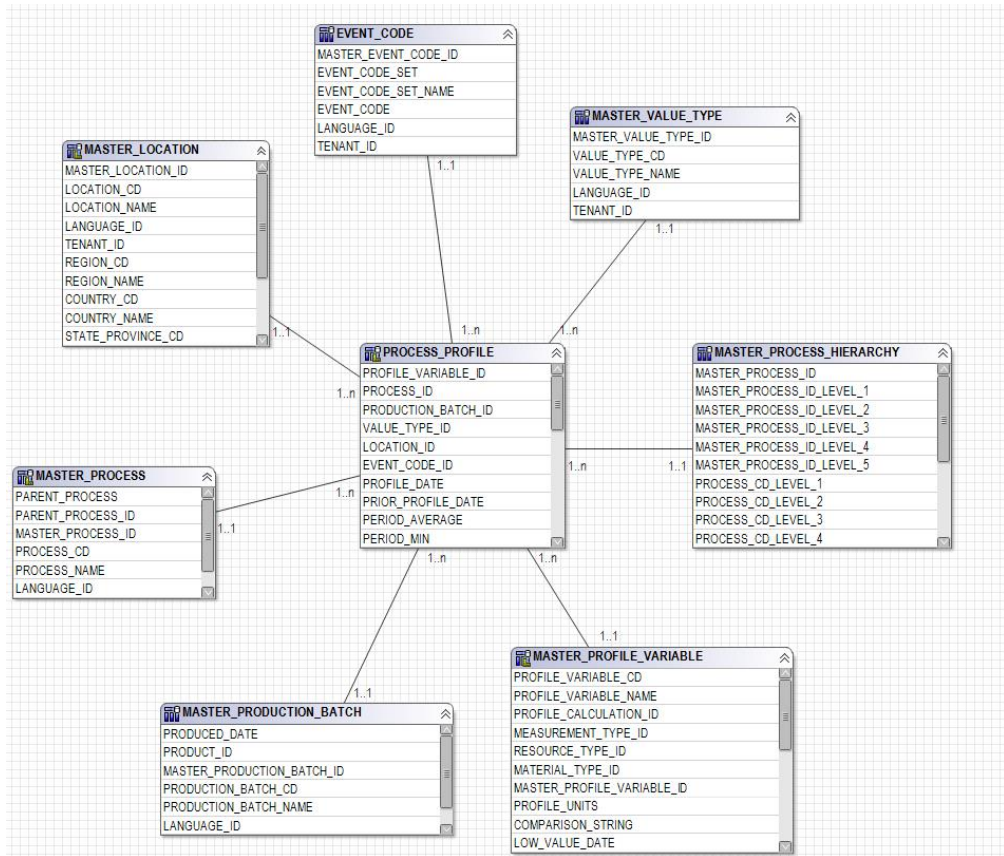


Figure 65. The process_profile star schema

The following diagram shows the star schema for the process_kpi table.

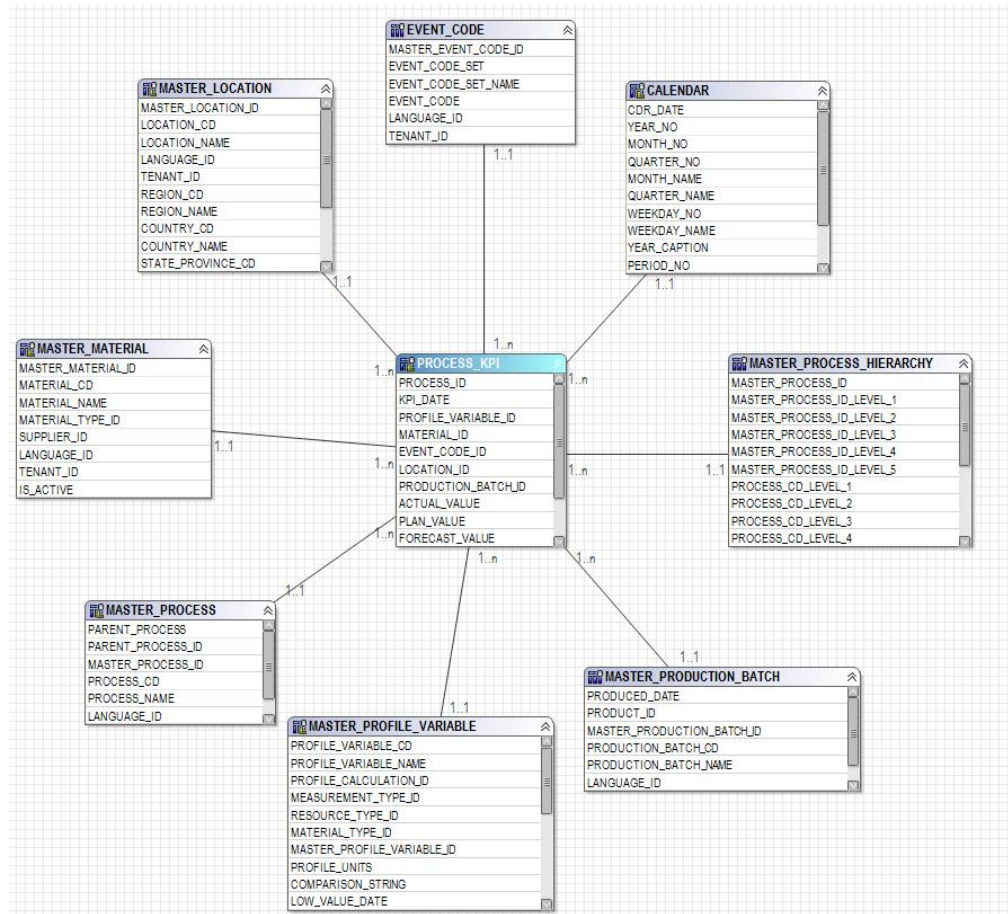


Figure 66. The process_kpi star schema

The following diagram shows the star schema for the lifetime_profile table.

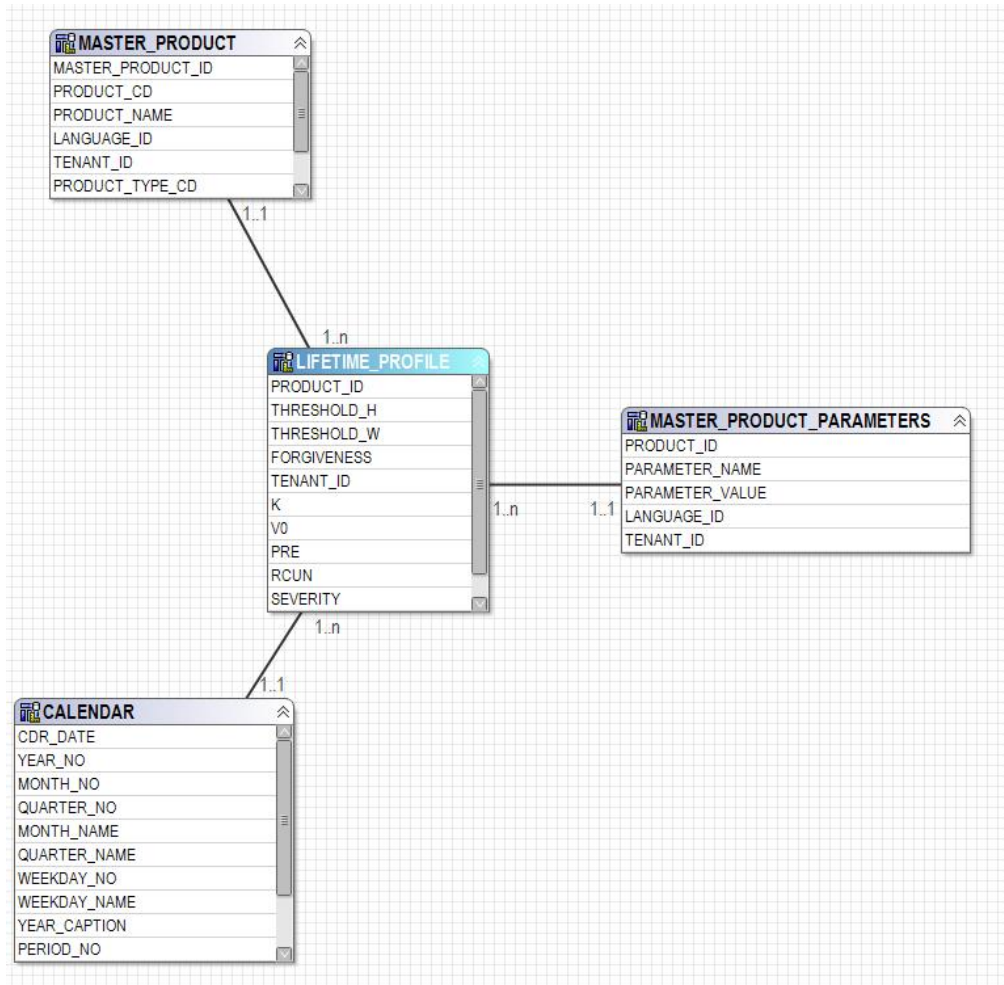


Figure 67. The `lifetime_profile` star schema

The following diagram shows the star schema for the `lifetime_kpi` table.

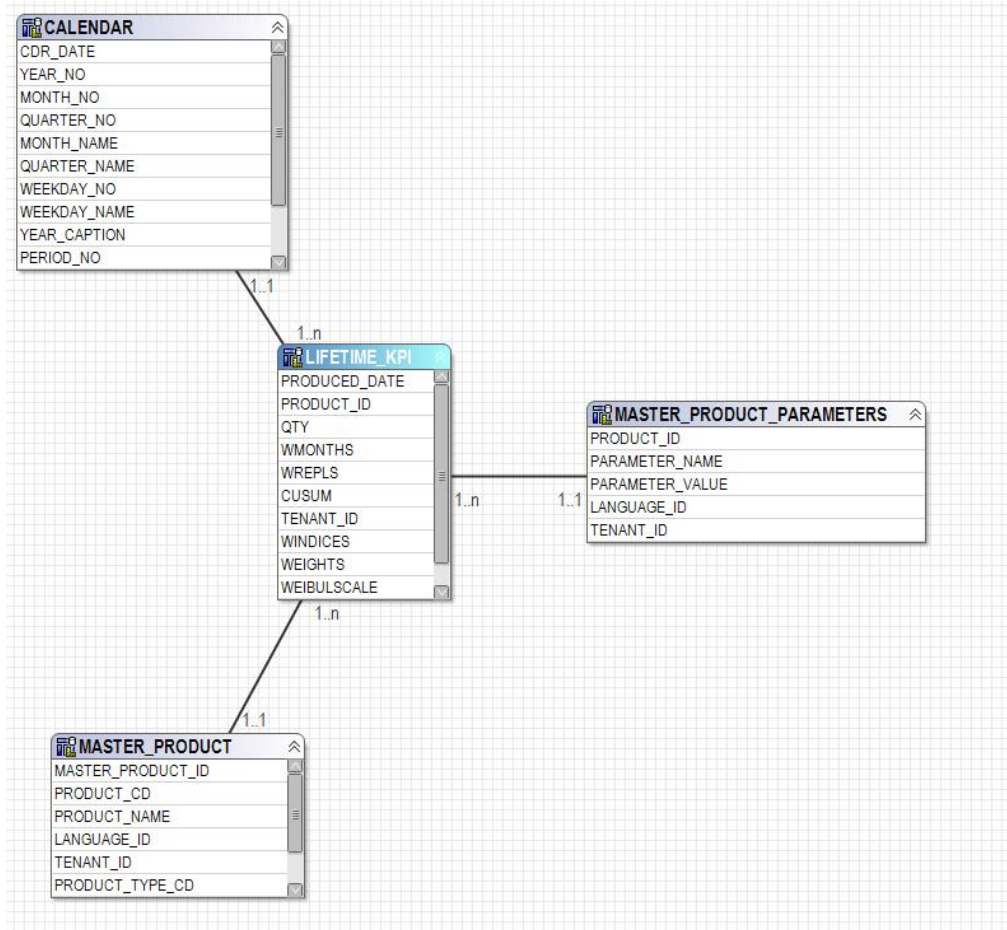


Figure 68. The `lifetime_kpi` star schema

The following diagram shows the star schema for the `maintenance_trends` table.

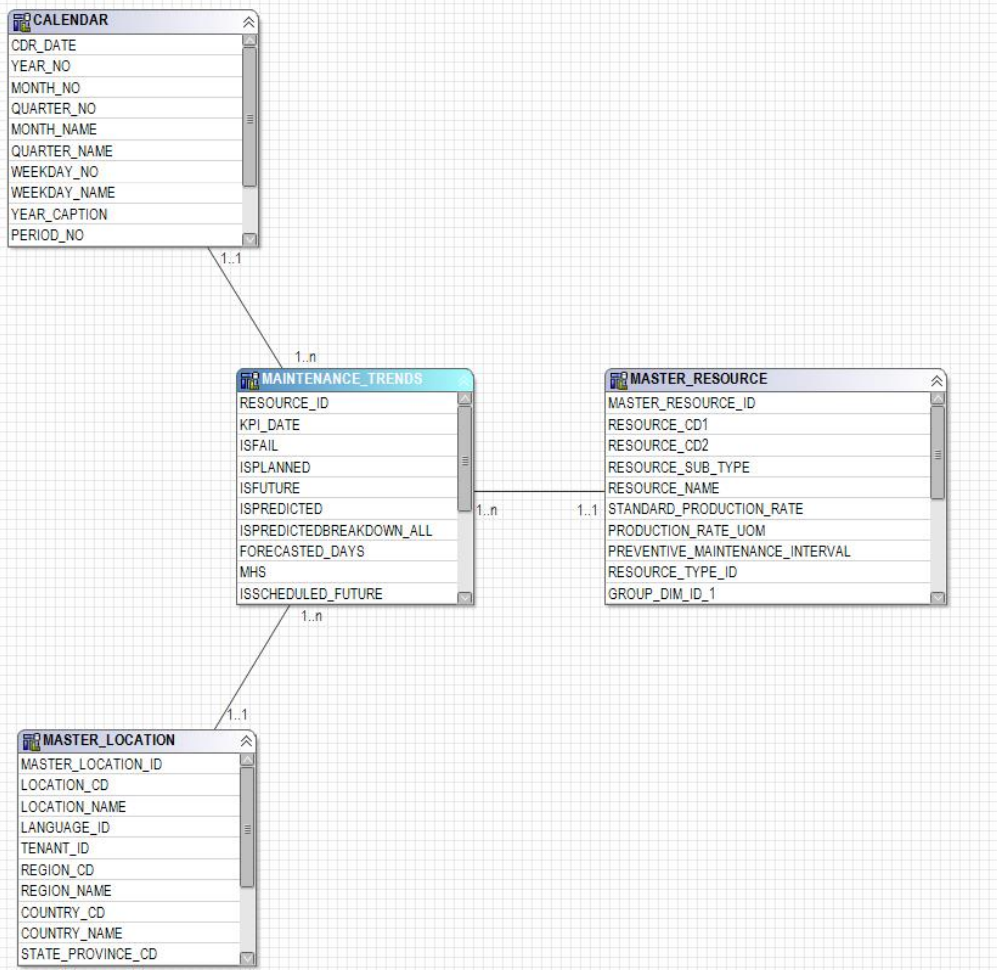


Figure 69. The maintenance_trends star schema

The following diagram shows the star schema for the product_kpi table.

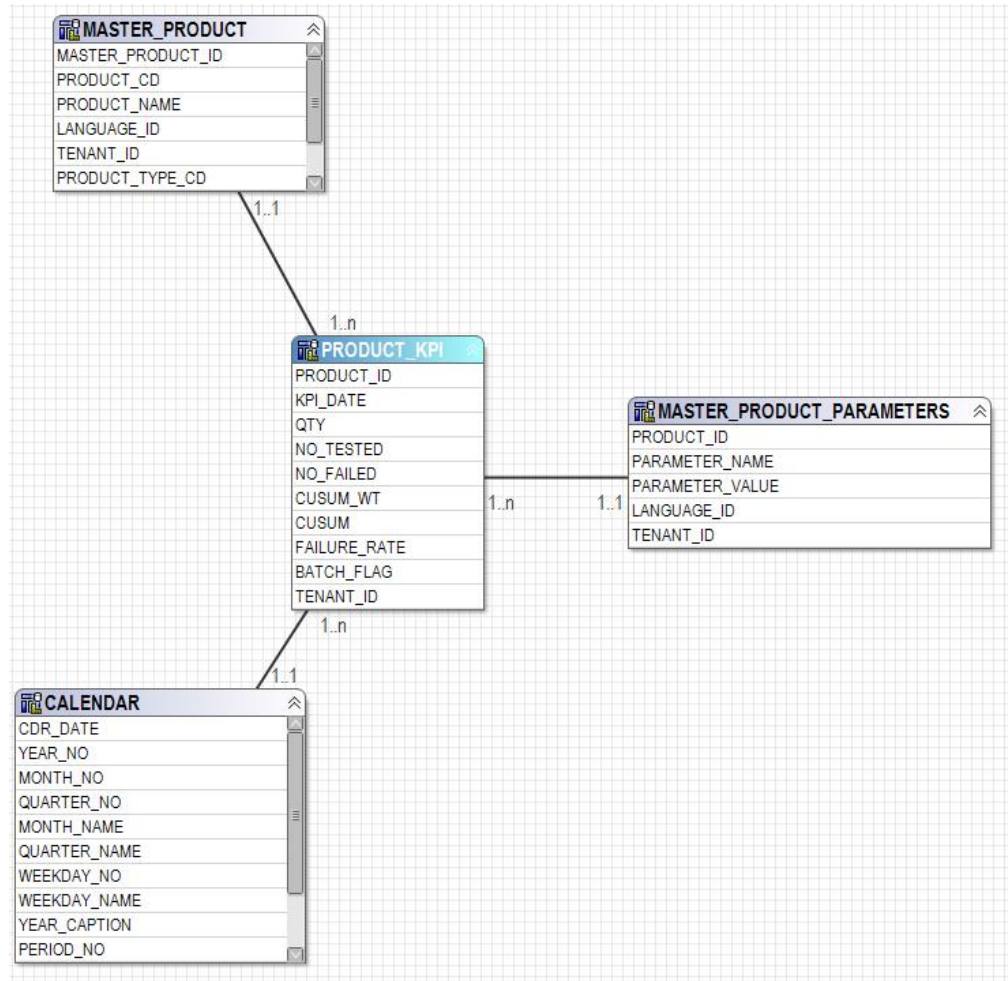


Figure 70. The product_kpi star schema

The following diagram shows the star schema for the product_profile table.

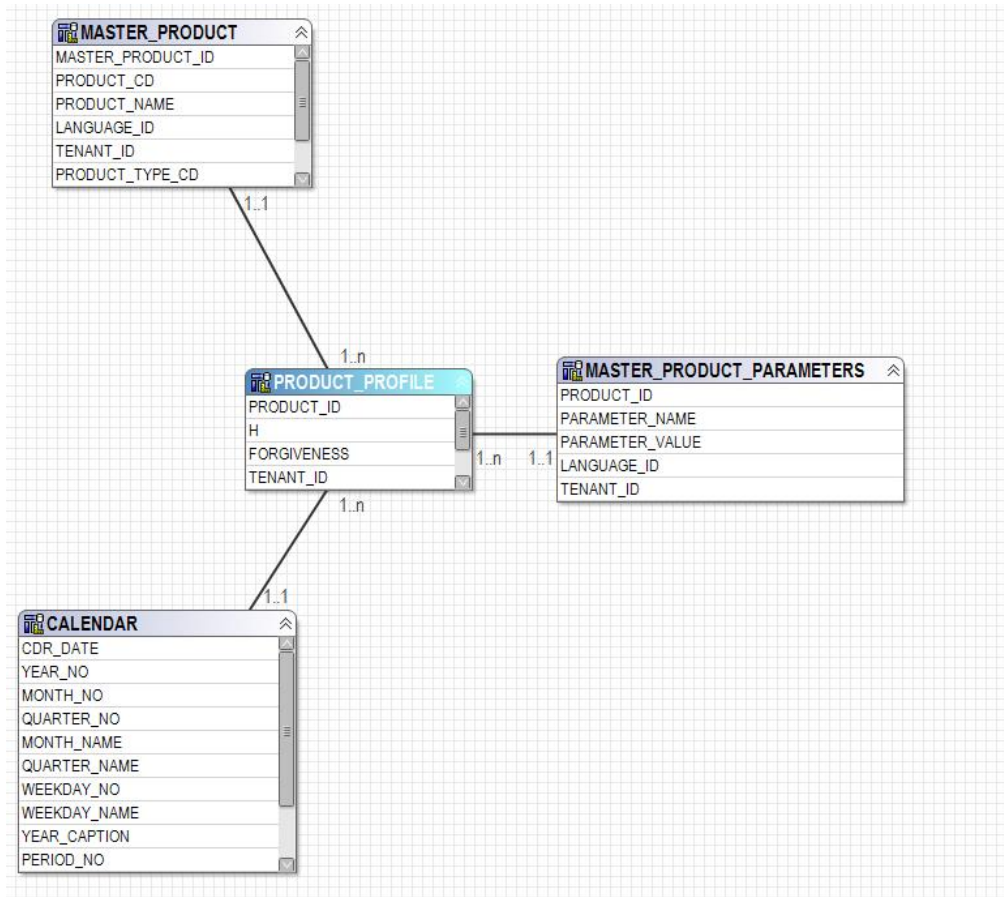


Figure 71. The product_profile star schema

The following diagram shows the star schema for the service table.

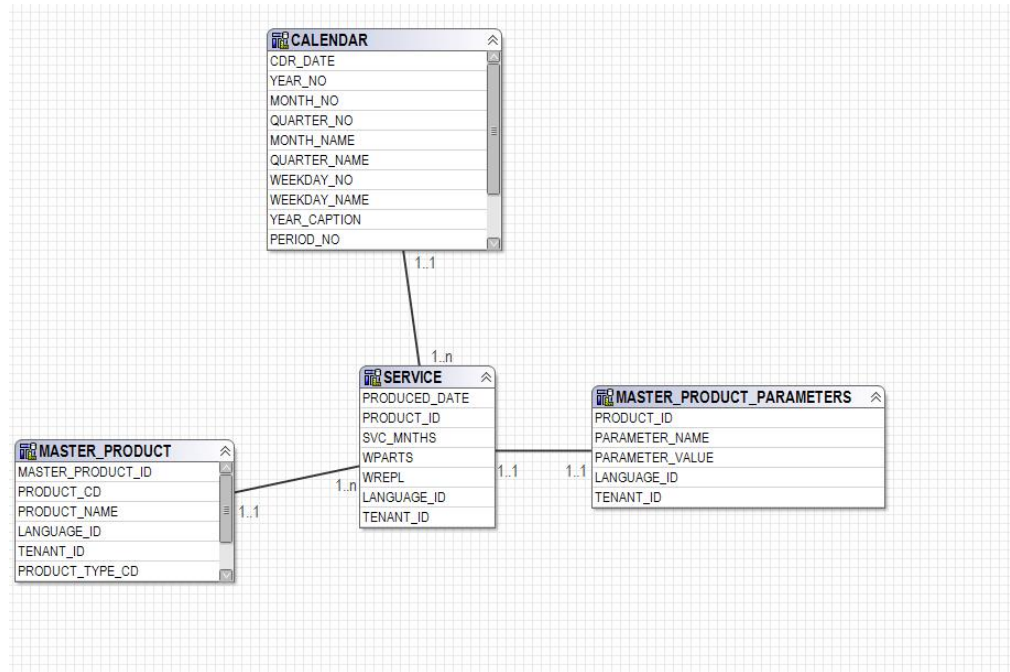


Figure 72. The service star schema

The following diagram shows the star schema for the parametric_kpi table.

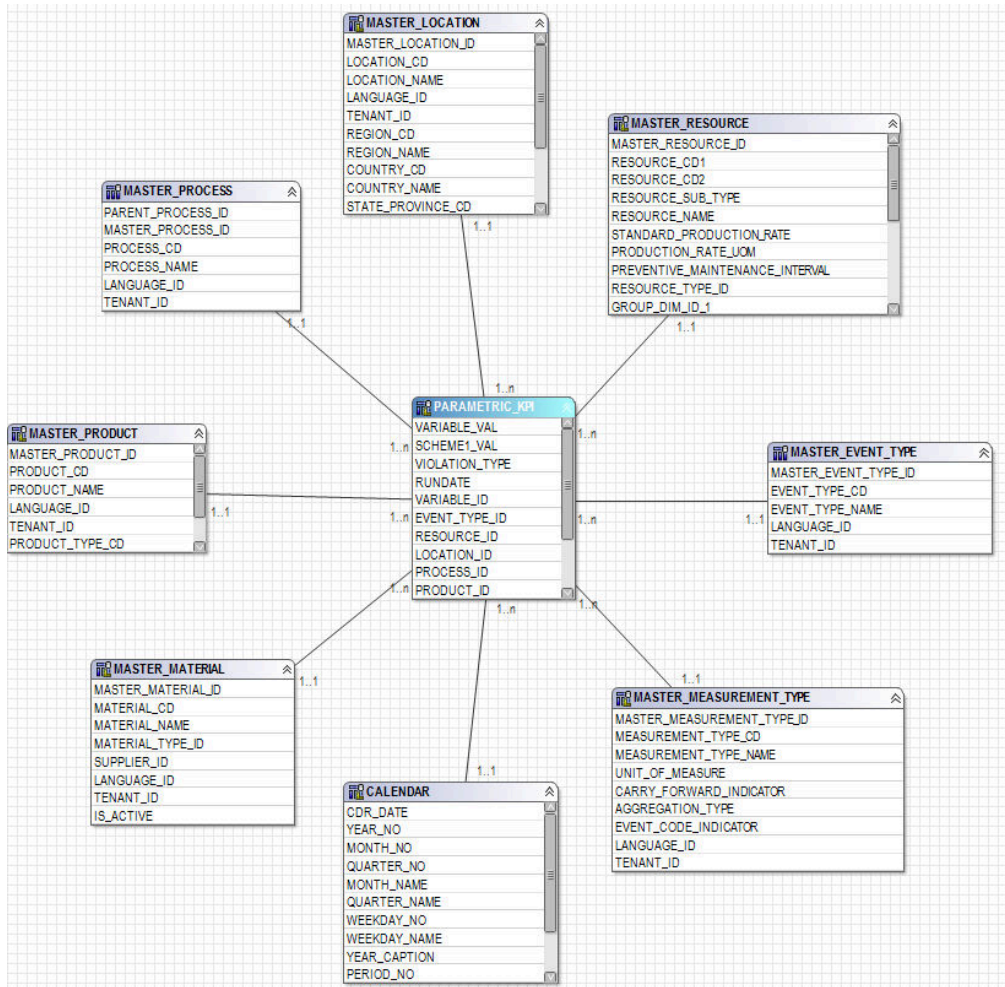


Figure 73. The `parametric_kpi` star schema

The following diagram shows the star schema for the `parametric_profile` table.

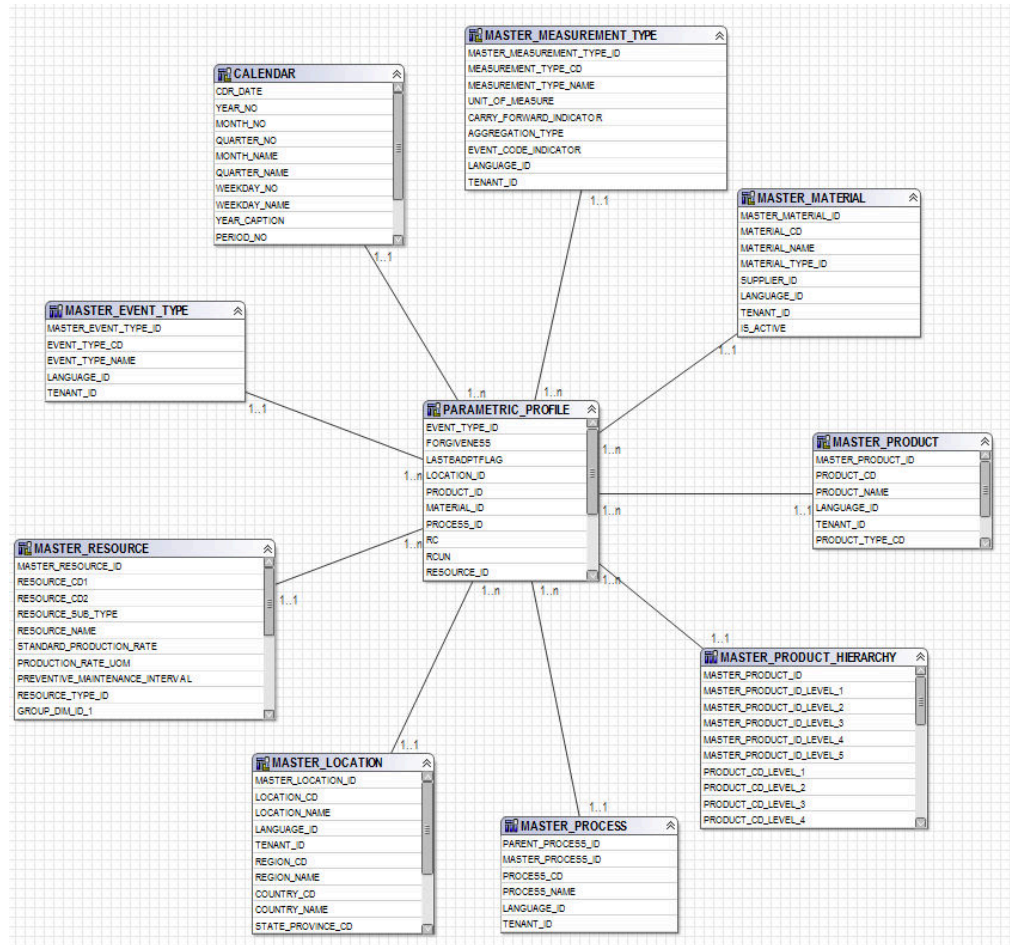


Figure 74. The parametric_profile star schema

The following diagram shows the star schema for the `profile_parameter` table.

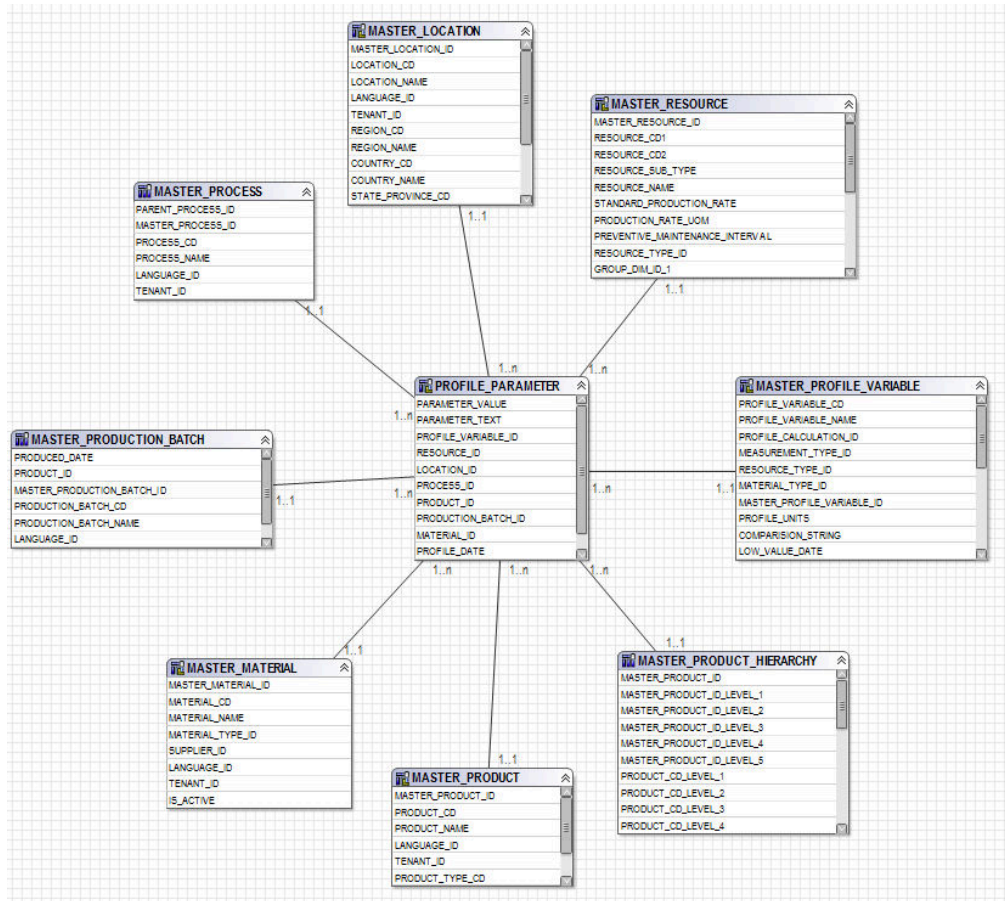


Figure 75. The profile_parameter star schema

IBM Cognos Framework Manager model logical layer

The logical layer contains query subjects that draw data from the database query subjects and present it in a more consumable format.

Attributes are renamed to eliminate underscores and to use sentence case. In some cases, physical entities are combined into one query subject. In particular, the following snowflake dimensions are combined together to accommodate master data reporting and avoid cross product result sets:

- Query subject Profile Variable contains attributes of profile_variable, measurement_type, profile_calculation, resource_type (profile_variable), and material_type (profile_variable).
- Query subject Material contains attributes of material, supplier and material_type.
- Query subject Production Batch contains attributes of production_batch and product
- Query subject Related Batch contains attributes of production_batch, batch_batch and production_batch (related).
- Query subject Resource contains attributes of resource, resource_type, location (resource), and group_dim_1 to 5.

- Query subject Event Observation contains attributes of event, event_observation, and event_resource.

The query subjects are organized into a folder for dimensions and a separate namespace for each logical fact. The fact query subjects contain additional calculated attributes that are included in dimensional layer measure dimensions.

IBM Cognos Framework Manager model dimensional layer

The dimensional layer contains the hierarchies and measure dimensions for publication to a package. Each dimension in the logical layer has a dimension in the dimensional layer with one or more hierarchies defined. The hierarchies usually include the caption field twice, once as a caption for the level, once as an attribute that can be used in report filters. All hierarchies are sorted.

Each measure dimension is in a separate namespace for the fact. Also included in the namespace are shortcuts to all the dimensions that have scope for that fact. Any dimension shortcut that is inside the namespace of the fact can also be consumed from outside of the namespace by IBM Cognos Business Intelligence reports.

Key Performance Indicator (KPI) tables include one measure with flexible aggregation. Based on the aggregation type in Profile Variable, the measure will either total the Actual Value or will calculate an average based on Sum of Actual Value / Sum of Measure Count. This requires that the data integration layer populates Measure Count with the actual number of observations for measures with aggregation type of Average and that it adds together measures that do not naturally seem to be additive, for example, temperature or pressure. The profile tables include a similar measure for flexible aggregation, with the addition of a check for Value Type = Actual.

IBM Cognos Framework Manager model security

No security is defined for the IBM Cognos Framework Manager model other than the provision for filtering by the tenant_id parameter on the physical layer. These query subject filters can be converted to security filters that are based on user IDs, allowing multi-tenant access to one database.

The Framework Manager model gives the ability to filter by the tenant_id parameter on the physical layer. As a preliminary measure to define security for the Framework Manager model, convert database query subject filters to security filters that are based on user IDs, allowing multi-tenant access to one database.

Query mode

The IBM Predictive Maintenance and Quality reports uses IBM Cognos Compatible Query Mode, which is the supported mode for all of the reports.

Using Compatible Query Mode to see real-time data

To see real-time data, you must ensure that caching is disabled in Dynamic Query mode and switch IBM Predictive Maintenance and Quality to use Compatible Query Mode.

Procedure

1. To turn off query reuse, open the CQEConfig.xml file that is in {IBM Cognos Install Directory}/configuration and edit the QueryEngine section by typing the following information.

```
<section name="QueryEngine">
  <!-- Description: queryReuse feature -->
  <!-- value="0" means disable the feature -->
  <!-- default is value="5" which means cache up to 5result sets per session -->
  <entry name=queryReuse" value="0"/>
  ...
</section>
```

2. Restart the IBM Cognos Business Intelligence server.
3. In IBM Cognos Administration, ensure that the data source defined for the IBM Predictive Maintenance and Quality database has native and JDBC connection definitions.
4. In IBM Framework Manager, select the project and change the **Query Mode** property to Compatible.
5. Publish the **IBMPMQ** package in Compatible mode by not selecting the check box to publish in Dynamic Query Mode when prompted.

Appendix D. IBM Predictive Maintenance and Quality Artifacts

IBM Predictive Maintenance and Quality (PMQ) artifacts contain the configuration files that provide connections to customer data, predictive models, rules, dashboards, reports, and external systems.

PMQ artifacts also contain sample data to aid in the understanding of how PMQ connects, manages, and analyzes data to produce business tools in the form of reports, dashboards, or maintenance work orders. These artifacts can be modified, as explained in this solution guide, for additional asset model requirements, event types, custom reports, or connections to other external data sources or systems of engagement.

Data model

The data model file name is `IBMPMQ.sql`. This DDL contains scripts to create all the tables that form the IBM Predictive Maintenance and Quality Master/Event/Profile Data mart. It contains stored procedures for the initial set up of Language and Tenant Data to perform basic operations required by Predictive Maintenance and Quality functions.

IBM InfoSphere Master Data Management Collaboration Server file

The IBM InfoSphere MDM Collaboration Server data model file name is `IBMPMQ.zip`. This is a company archive file that contains all the templates, reports, and data of the MDM CE data model specific to PMQ Master Data.

IBM Integration Bus and ESB artifacts

IBM Integration Bus (IIB) and Enterprise Service Bus (ESB) artifacts are provided.

IBM Integration Bus archive files

IBM Integration Bus archive files are shown in the following table:

Table 70. IBM Integration Bus archive files

SI. Number	BAR Files	Description
1.	PMQMasterDataLoad	onboard Master Data information on to the PMQ Data mart.
2.	PMQEventDataLoad	Onboard and process event Data information on to the PMQ Event Store Integrate with SPSS scoring services (sensor health score and integrated health score) and process score results
3.	PMQMaintenance	Perform data preparations and invoke SPSS Maintenance Job as per the schedule
4.	PMQTopNFailure	Perform data preparations and invoke SPSS TopN Failure Job as per the schedule

Table 70. IBM Integration Bus archive files (continued)

SI. Number	BAR Files	Description
5.	PMQQEWSInspection	Prepares data and invokes the QEWS algorithm to perform inspection early warning analysis and loads the results back to the Profile data Mart of PMQ.
6.	PMQQEWSWarranty	Gathers data from the Service tables of PMQ data mart and passes as input to the QEWSL analysis and loads the results to the Profile data mart of PMQ.
7.	PMQMaximoIntegration	load master data and workorder from Maximo in PMQ and also supports creation/updating of Maximo workorders
8.	PMQQEWSIntegration	provide integration support to call Inspection and Warranty flows as per the required sequence or schedule and to invoke SPSS warranty stream
9.	PMQModelTraining	Invoke SPSS Job for training SPSS streams for sensor health score and integrated health score

Supported JAR files

Supported JAR files are shown in the following table:

Table 71. Supported JAR files

SI. Number	JAR/Properties/XML files	Description
1.	foundation-engine-api-1.5.0.0-SNAPSHOT.jar	APIs provided by Analytic Solution Foundation 1.5
2.	foundation-engine-core-1.5.0.0-SNAPSHOT.jar	Implementation jar of Analytics Solution Foundation 1.5
3.	org.apache.commons-collections-3.2.1.jar	This jar provides utility methods for most of the collection interfaces.
4.	commons-io-2.4.jar	This is a library of utilities to assist with developing IO functionality
5.	org.apache.commons-lang-2.4.jar	Provides a host of helper utilities for the java.lang API, notably String manipulation methods
6.	commons-pool-1.6.jar	This open source software library provides an object-pooling API and a number of object pool implementations.
7.	hamcrest-core-1.3.jar	Provides a library of matcher objects allowing 'match' rules to be defined declaratively, to be used in other frameworks.
8.	log4j-1.2.16.jar	Serves methods for logging purpose.
9.	icu4j.53.1.jar	Serves for internationalization
10.	pmq-foundation.jar	PMQ custom calculations over what is supported by Foundation
11.	ews.jar	The early warning system java module to analyze inspection and warranty usecases.

Supported property files and XML files

Supported property files and XML files are shown in the following table:

Table 72. Supported property files and XML files

Sl. No.	JAR / Properties / XML files
1	SetPerm.sh - Used to set 755 on the folder structure containing Warranty and Inspection charts
2	credentials.properties - Used to store SPSS credentials and joblocation urls
3	loc.properties - This is a properties file which maintains the location information of where the outputs to Warranty and Inspection is to be rendered.
4	log4j.properties - This is to set the logging levels and paths for the logs to be persisted.
5	orchestration_definition.xsd - foundation orchestration schema
6	solution_definition.xsd - foundation solution schema
7	<ul style="list-style-type: none"> • PMQ_orchestration_definition_inspection.xml • PMQ_orchestration_definition_maintenance.xml • PMQ_orchestration_definition_measurement.xml • PMQ_orchestration_definition_topnfailure.xml • PMQ_orchestration_definition_warranty.xml <p>These Foundation specific orchestration XML contains the orchestration mapping definitions to carry out the sequence of adapter calls to fulfill an operation. We have a separate XML for each of the use case/event type.</p>
8	PMQ_solution_definition.xml. This Foundation specific XML contains the table definitions and the relations to carry the DML and DDL operations.
13	<ul style="list-style-type: none"> • PMQEventLoad.properties • PMQMaintenance.properties • PMQMaximoIntegration.properties • PMQModelTraining.properties • PMQQEWSIntegration.properties • PMQTopNFailure.properties <p>These properties files will contain the webservice endpoint urls and are used to override the bar files with the correct endpoint urls as per customer needs</p>
14	Queues.txt - Contains all supporting queue definitions' and this is executed to create queues

Sample master data, event data, and QEWS data files

Sample master data files, event data files, and QEWS data files are provided.

The sample master data files are shown in the following list:

- language_upsert.csv
- tenant_upsert.csv
- event_code_upsert.csv
- event_type_upsert.csv
- group_dim_upsert.csv
- location_upsert.csv

- material_type_upsert.csv
- measurement_type_upsert.csv
- observation_lookup_upsert.csv
- process_upsert.csv
- product_upsert.csv
- profile_calculation_upsert.csv
- resource_type_upsert.csv
- source_system_upsert.csv
- supplier_upsert.csv
- value_type_upsert.csv
- material_upsert.csv
- production_batch_upsert.csv
- profile_variable_upsert.csv
- resource_upsert.csv

The sample event data files are shown in the following list:

- event_observation_maintenance_training.csv
- event_observation_maintenance_training_recommendation.csv
- event_observation_sensor_training.csv
- event_observation_process_material.csv
- event_observation_spc.csv
- event_observation_sensor.csv

The QEWS data files are shown in the following list:

- parameter_upsert.csv
- resource_production_batch_upsert.csv
- batchdata_inspection.csv
- event_observation_warranty.csv
- qewsrundate.txt

IBM SPSS artifacts

IBM SPSS streams and jobs are provided as artifacts.

Warranty - Streams and Jobs

Warranty artifacts are shown in the following table:

Table 73. Warranty - Streams and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_QEWSL	IBMPMQ_QEWSL_WARR.str	The manufacturing or production warranty stream built to do a ETL sort of processing. No modeling activity involved here
	IBMPMQ_QEWSL_JOB	CaDS job used to invoke IBMPMQ_QEWSL_WARR.str for the manufacturing (MFG) or production (PROD)use cases

Table 73. Warranty - Streams and Jobs (continued)

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
	IBMPMQ_QEWSL_SALES.str	CaDS job used to invoke IBMPMQ_QEWSL_JOB for the sales (SALES) use case
	IBMPMQ_QEWSL_SALES_JOB	CaDS job used to invoke IBMPMQ_QEWSL_SALES.str for the SALES use cases

Maintenance - Stream and Jobs

Maintenance artifacts are shown in the following table:

Table 74. Maintenance - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_MAINTENANCE_ANALYTICS	MAINTENANCE.str	Main stream in maintenance to identify and forecast the forecasted days to next maintenance and calculate the Maintenance Health score value .
	MAINTENANCE_DAILY.str	Gives the Maintains details for a specific day
	MAINTENANCE_RECOMMENDATIONS.str	The ADM Stream to give the Maintenance recommendations
	IBMPMQ_MAINTENANCE_ANALYTICS_JOB	CaDS job used to invoke MAINTENANCE.str, MAINTENANCE_DAILY.str, MAINTENANCE_RECOMMENDATIONS.str, and IBMPMQ_MAINTENANCE_ANALYTICS_JOB

TopN failure predictors - Stream and Jobs

TopN failure predictors artifacts are shown in the following table:

Table 75. TopN failure predictors - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_TOP_FAILURE_PREDICTORS	TopN_MODEL.str	Modeling stream to extract and store the PMML giving the predictor importance of various configured parameters in predicting a resource's failure.

Table 75. TopN failure predictors - Stream and Jobs (continued)

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
	TopN_XML.str	This stream uses the PMML generated by the TopN_MODEL.str stream and extracts the necessary information from it and does essential transformation such that the output can be consumed by Cognos
	IBMPMQ_TOP_FAILURE_PREDICTORS_JOB	CaDS job used to invoke the TopN_MODEL.str and the TopN_XML.str streams
	TOPN_EVENTS.str	Create csv with of the Top N data in a format that can be loaded into the PMQ event table using the IIB flows

SENSOR-based health analytics - Stream and Jobs

SENSOR-based health analytics artifacts are shown in the following table:

Table 76. SENSOR-based health analytics - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_SENSOR_ANALYTICS	SENSOR_HEALTH_DATA_PREP.str	A Data preparation stream which retrieves the data from IBM PMQ tables and prepares the data to be used in the modeling, the eligible data is exported to a csv file for the modeling
	SENSOR_HEALTH_COMBINED.str	The combined stream helps in training the models and also refreshes them for the scoring service
	SENSOR_HEALTH_ANALYTICS_JOB	CaDS job used to invoke the SENSOR_HEALTH_COMBINED.str stream
	IBMPMQ_SENSOR_ANALYTICS.str	This stream is auto generated when a training happens and for the real time scoring -- SENSOR_HEALTH_SCORE service configured to be used

Integrated analytics - Stream and Jobs

Integrated analytics artifacts are shown in the following table:

Table 77. Integrated analytics - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_INTEGRATED_FEATURE_BASED_ANALYTICS	INTEGRATION_FBA_DATA_PREP.str	A Data preparation stream which retrieves the data from IBM PMQ tables and prepares the data to be used in the modeling, the eligible data is exported to a csv file for the modeling
	INTEGRATION_FBA_IHS_T.str	This stream helps in training the health score models and also refreshes them for the scoring service
	INTEGRATION_FBA_IFDM_T.str	This stream helps in training the forecasted days to maintenance model and also refreshes the model for the scoring service
	IBMPMQ_INTEGRATED_FEATURE_BASED_ANALYTICS	CaDS job used to invoke the INTEGRATION_FBA_DATA_PREP.str , INTEGRATION_FBA_IHS_T.str, and INTEGRATION_FBA_IFDM_T.str streams
	INTEGRATION_FBA_IHS.str	This stream is auto generated when a training happens, and is invoked for real time health score prediction within the INTEGRATED_FBA.str stream
	INTEGRATION_FBA_IFDM.str	This stream is auto generated when a training happens, and is invoked for real time forecasted days to maintenance prediction within the INTEGRATED_FBA.str stream

Table 77. Integrated analytics - Stream and Jobs (continued)

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
	INTEGRATED_FBA.str	This is ADM published stream which invokes the health score and days to maintenance models, and gives a final recommendation based on business rules. It is configured for real time scoring - INTEGRATED_FBA service is configured for scoring services.

Feature-based analytics - Stream and Jobs

Feature-based analytics artifacts are shown in the following table:

Table 78. Feature-based analytics - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_SENSOR_ FEATURE_BASED ANALYTICS	SENSOR_FBA_DATA_ PREP.str	A Data preparation stream which retrieves the data from IBM PMQ tables and prepares the data to be used in the modeling, the eligible data is exported to a csv file for the modeling
	SENSOR_FBA_FHS_T.str	This stream helps in training the health score models and also refreshes them for the scoring service
	SENSOR_FBA_FFDM_T.str	This stream helps in training the forecasted days to maintenance model and also refreshes the model for the scoring service
	IBMPMQ_SENSOR_ FEATURE_BASED_ ANALYTICS	CaDS job used to invoke the SENSOR_FBA_DATA_ PREP.str, SENSOR_FBA_ FHS_T.str, and SENSOR_FBA_ FFDM_T.str streams
	SENSOR_ FBA_FHS.str	This stream is auto generated when a training happens, and is invoked for real time health score prediction within the INTEGRATED_FBA.str stream

Table 78. Feature-based analytics - Stream and Jobs (continued)

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
	SENSOR_FBA_FFDM.str	This stream is auto generated when a training happens, and is invoked for real time forecasted days to maintenance prediction within the FBA.str stream
	FBA.str	This is ADM published stream which invokes the health score and days to maintenance models, and gives a final recommendation based on business rules. It is configured for real time scoring - FBA service configured to be used.

Features for the energy and utility industry - Stream and Jobs

Energy and utility industry artifacts are shown in the following table:

Table 79. Features for the energy and utilities industry - Stream and Jobs

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
IBMPMQ_ARMOR_ANALYTICS	IBMPMQ_ARMOR_DTCA_CABLE_FEATURES	CaDS job used to invoke the IBMPMQ_FEATURES_DTCA.str and IBMPMQ_FEATURES_CABLE.str to generate features for distribution transformer and cable assets respectively.
	IBMPMQ_FEATURES_DTCA.str	This stream helps to generate the csv file with Distribution Transformer specific overload features, as well as current aging measure. The csv file is used in Feature-Based Analytics modeling.
	IBMPMQ_FEATURES_CABLE.str	This stream helps to generate the csv file with Cable specific overload features, which is used in FBA modeling.
	IBMPMQ_ARMOR_DTPA	CaDS job used to invoke the IBMPMQ_FEATURES_DTPA.str stream to generate projected aging for distribution transformer assets.

Table 79. Features for the energy and utilities industry - Stream and Jobs (continued)

.pes file	Modeler Stream / ADM Streams / CaDS jobs	Description
	IBMPMQ_FEATURES _DTPA.str	This stream helps to generate the csv file with distribution transformer projected aging for user-specified future years (based on start year, number of years, and interval size), and degradation factor.

IBM Cognos Business Intelligence Artifacts

An IBM Framework Manager model, and a compressed file that contains reports and dashboards is provided.

Framework Manager model

The Framework Manager model is described in the following table:

Table 80. Framework Manager model

SI. Number	FM Model	Purpose
1.	IBMPMQ	<p>IBM Predictive Maintenance and Quality uses IBM Cognos Framework Manager to model the metadata for reports. IBM Cognos Framework Manager is a metadata modeling tool that drives query generation for IBM Cognos software.</p> <p>A model is a collection of metadata that includes physical information and business information for one or more data sources. IBM Cognos software enables performance management on normalized and denormalized relational data sources and a variety of OLAP data sources.</p>

Site Overview dashboard

The Site Overview dashboard is described in the following table:

Table 81. Site Overview dashboard

SI. Number	Report/Dashboard	Purpose
1.	Overview	<p>Provides a high-level summary of the health of all of your assets at all sites, it shows the key performance indicators (KPIs) with the greatest impact.</p> <p>You can change the detail that is displayed by selecting items from the list boxes. For example, you can change the date and the equipment type.</p>
2.	Top 10 Contributors	Identifies the equipment, locations, and operators responsible for the most failures.

Table 81. Site Overview dashboard (continued)

Sl. Number	Report/Dashboard	Purpose
3.	KPI Trending	<p>You can select multiple key performance indicators (KPIs) to be plotted side-by-side in a line chart.</p> <p>You can identify correlations between the KPIs and see whether there is any lagging behavior.</p> <p>For example, if there is a spike in one KPI, how long does it take to impact the other KPIs?</p>
4.	Actual vs Plan	<p>You can monitor how closely the metrics track against the plan.</p> <p>Variances are highlighted.</p>
5.	Equipment Listing	<p>The health score for a site is derived from the lower-level scores from each piece of equipment in the site.</p> <p>This report shows you all the pieces of equipment on the site and the health scores and relevant KPIs for that equipment.</p>
6.	Equipment Outliers	<p>Lists the equipment (or assets) that are performing outside of allowable limits. The measures that are shown differ depending on the equipment, but examples are operating temperature, lateral strain, hydraulic pressure, average value, last value, and control limits.</p>
7.	Recommended actions	<p>A summary of all recommended actions for each piece of equipment, for the health score measurement.</p>

Equipment Reports dashboard

The Equipment Reports dashboard is described in the following table:

Table 82. Equipment Reports dashboard

Sl. Number	Report/Dashboard	Purpose
1.	Equipment Profile	<p>A detailed report that shows everything that is known about a piece of equipment: how it is performing today and how it performed in the past.</p>
2.	Equipment Control Chart	<p>Shows the upper and lower control limits and the average limits for selected measures.</p>
3.	Equipment Run Chart	<p>Shows the measures for a particular piece of equipment.</p>
4.	Equipment Outliers	<p>Shows detailed measures for a piece of equipment that shows anomalies.</p>

Table 82. Equipment Reports dashboard (continued)

Sl. Number	Report/Dashboard	Purpose
5.	Event Type History	Lists the events for a device.

Product Quality dashboard

The Product Quality dashboard is described in the following table:

Table 83. Product Quality dashboard

Sl. Number	Report/Dashboard	Purpose
1.	Defect Analysis	Shows product defects and inspection rates.
2.	Inspection Rate Analysis	Examines the relationship between inspections and defects over time to find the optimal rate of inspection.
3.	Material Usage By Process	Provides an overview of material usage in the production processes.

SPC reports

SPC reports are described in the following table:

Table 84. SPC reports

Sl. Number	Report/Dashboard	Purpose
1.	SPC - Histogram	This report allows a visual interpretation of data by indicating the number of data points (events) that lie within a range of values, called a class or a bin. The frequency of the data that falls in each bin is depicted by the use of a bar.
2.	SPC - X Bar and S / R Charts	To track instantaneous variations and to evaluate the stability of the variability with in the process for smaller sample sizes (R Chart) and for bigger sample sizes (S Chart)

Other reports

Other reports are described in the following table:

Table 85. Other reports

Sl. Number	Reports/Dashboard	Purpose
1.	Advance KPI Trend Report	This chart compares multiple key performance indicators (KPIs) across multiple resources. You can use this chart to analyze variations in a resource against a set of profiles.
2.	Material Usage by Production Batch	This report provides an overview of material usage by production batch. By correlating production batches with defects to material usage by production batch, you can begin to trace the impact of defective materials.
3.	Audit Report	Shows the counts of rows in the major master data tables.

Drill Through reports from the Audit report

The following table lists Drill Through reports from the Audit report.

Table 86. Drill Through reports from the Audit report

Sl. Number	Reports/Dashboard	Purpose
1.	Resource List	Lists the resources by resource type.
2.	Profile Variables	Lists all measures and key performance indicators that are being tracked in daily profiles and historical snapshots.
3.	Process List	Lists all production processes.
4.	Material List	Lists materials that are used in the production process.
5.	Production Batch List	Lists production batches.
6.	Material Usage By Production Batch	This report provides an overview of material usage by production batch. By correlating production batches with defects to material usage by production batch, the impact of defective materials can begin to be traced.
7.	Measurement Type List	Lists measurement types. For each measurement type, the report shows unit of measure and aggregation type.

Maintenance dashboard and Top N Failure reports

The Maintenance dashboard and Top N Failure reports are described in the following table:

Table 87. Maintenance dashboard and Top N Failure reports

Sl. Number	Reports/Dashboard	Purpose
1.	Maintenance Overview Dashboard	This dashboard provides an overview of health score for the last current day in the record. Along with maintenance health score, report also shows a comparative view with sensor health score and integrated health score.
2.	Maintenance Advance Sorting Report	This chart displays the same measures as the main report (Maintenance Overview Dashboard) in a tabular format. Users can sort on a column by clicking the column header.
3.	Maintenance Health Score and Failure Detail Report	This report will help user to see the historical and forecasted health scores of a machine along with Historical Breakdown, Forecasted Breakdown, Planned Maintenance Schedules.
4.	Top N Failure Report	The plot shows the UNSIGNED predictor importance, indicating the absolute importance of any predictor in predicting a failure or non-failure condition.

QEWS quality dashboards and reports

QEWS quality dashboards and reports are described in the following table:

Table 88. Inspection and Warranty reports

Sl. Number	Reports/Dashboard	Purpose
1.	Quality dashboard - Inspection	This dashboard provides an overview of the state of products at a selected run date.
2.	Quality dashboard - Inspection Detail History	This dashboard provides details about the state of products and the various threshold values for a selected product category at a selected run date.
3.	QEWS - Inspection Chart	This chart reports the failure rates and CUSUM values for a specific product type and product code over a time period.

Table 88. Inspection and Warranty reports (continued)

Sl. Number	Reports/Dashboard	Purpose
4.	Quality dashboard - Warranty	This dashboard provides an overview of the state of products at a selected run date.
5.	Quality dashboard - Warranty Detail History	This dashboard provides details about the state of products and the various threshold values for a selected product category at a selected run date.
6.	QEWSL - Warranty Chart	This chart reports the replacement rates for a specific product type and product code over a time period.
7.	Quality dashboard - Parametric	This dashboard provides an overview of the state of products at a selected run date for a variable.
8.	Quality dashboard - Parametric Detail History	This dashboard provides details about the state of products and the various threshold values for a selected product category, at a selected run date for a variable.
9.	QEWSV - Parametric chart	<p>This report is used for monitoring variable-type data and CUSUM values that are obtained from the QEWSV batch along with threshold levels.</p> <p>The report is designed to support five different validation types: Material Validation, Process-Resource Validation, Production Batch Validation, Resource Health Check, and Location Suitability.</p>

Appendix E. Troubleshooting

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem.

Review the following table to help you or customer support resolve a problem.

Table 89. Troubleshooting actions and descriptions

Actions	Description
A product fix might be available to resolve your problem.	Apply all known fix packs, or service levels, or program temporary fixes (PTF).
Look up error messages by selecting the product from the IBM Support Portal, and then typing the error message code into the Search support box (http://www.ibm.com/support/entry/portal/).	Error messages give important information to help you identify the component that is causing the problem.
Reproduce the problem to ensure that it is not just a simple error.	If samples are available with the product, you might try to reproduce the problem by using the sample data.
Ensure that the installation successfully finished.	The installation location must contain the appropriate file structure and the file permissions. For example, if the product requires write access to log files, ensure that the directory has the correct permission.
Review all relevant documentation, including release notes, technotes, and proven practices documentation.	Search the IBM knowledge bases to determine whether your problem is known, has a workaround, or if it is already resolved and documented.
Review recent changes in your computing environment.	Sometimes installing new software might cause compatibility issues.

If the items on the checklist did not guide you to a resolution, you might need to collect diagnostic data. This data is necessary for an IBM technical-support representative to effectively troubleshoot and assist you in resolving the problem. You can also collect diagnostic data and analyze it yourself.

Troubleshooting resources

Troubleshooting resources are sources of information that can help you resolve a problem that you are having with an IBM product.

Support Portal

The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services.

The IBM Support Portal lets you access all the IBM support resources from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution. Familiarize yourself

with the IBM Support Portal by viewing the demo videos (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos).

Find the content that you need by selecting your products from the IBM Support Portal (<http://www.ibm.com/support/entry/portal/>).

Gathering information

Before contacting IBM Support, you will need to collect diagnostic data (system information, symptoms, log files, traces, and so on) that is required to resolve a problem. Gathering this information will help to familiarize you with the troubleshooting process and save you time

Service requests

Service requests are also known as Problem Management Reports (PMRs). Several methods exist to submit diagnostic information to IBM Software Technical Support.

To open a PMR or to exchange information with technical support, view the IBM Software Support Exchanging information with Technical Support page (<http://www.ibm.com/software/support/exchangeinfo.html>).

Fix Central

Fix Central provides fixes and updates for your system's software, hardware, and operating system.

Use the pull-down menu to navigate to your product fixes on Fix Central (<http://www-947.ibm.com/systems/support/fixes/en/fixcentral/help/getstarted.html>). You may also want to view Fix Central help.

Knowledge bases

You can find solutions to problems by searching IBM knowledge bases.

You can use the IBM masthead search by typing your search string into the Search field at the top of any [ibm.com](http://www.ibm.com) page.

IBM Redbooks

IBM Redbooks[®] are developed and published by IBM's International Technical Support Organization, the ITSO.

IBM Redbooks (<http://www.redbooks.ibm.com/>) provide in-depth guidance about such topics as installation and configuration and solution implementation.

IBM developerWorks

IBM developerWorks provides verified technical information in specific technology environments.

As a troubleshooting resource, developerWorks provides easy access to the top ten most popular practices for Business analytics, in addition to videos and other information: developerWorks for Business analytics (<http://www.ibm.com/developerworks/analytics/practices.html>).

Software support and RSS feeds

IBM Software Support RSS feeds are a quick, easy, and lightweight format for monitoring new content added to websites.

After you download an RSS reader or browser plug-in, you can subscribe to IBM product feeds at IBM Software Support RSS feeds (<https://www.ibm.com/software/support/rss/>).

Log files

Log files can help you troubleshoot problems by recording the activities that take place when you work with a product.

IBM Integration Bus log files

Errors that occur within IBM Integration Bus message flows are written to error logs in the following folder: /error. The location of this folder is determined by the **MQSI_FILENODES_ROOT_DIRECTORY** environment variable during the installation process.

Errors for message flows are as follows:

Master data flows

Rejected records are written to *input_filename_error.csv*

Errors are logged in *input_filename_error.txt*

Event flow - MultiRowEventLoad

Rejected records are written to *input_filename_error.csv*

Errors are logged in *input_filename_error.txt*

Event flow - StdEventLoad

Failed event messages are written to the error queue `PMQ.EVENT.ERROR`

Errors are logged in *EventError.txt*

PMQIntegration flow

Failed event request and web service fault messages are written to the error queue: `PMQ.INTEGRATION.ERROR`

Errors are logged in *IntegrationError.txt*

Maximo flow - Maximomasterdataasset, Maximomasterdataclassification, Maximomasterdatalocation

Rejected records are written to *input_filename_error.xml*

Errors are logged in *input_filename_error.txt*

Maximo flow - WorkorderCreation

Failed Maximo requests and web service fault message are written to the error queue: `PMQ.MAXIMO.ERROR`

Log files generated during the installation process

Errors that occur during the prerequisite checks that happen during the installation process are written to the following location on the node where installation is taking place:

```
/var/IBMPMQ/PreReq.log
```

The following errors can be reported:

Error, Can't proceed as the user is a Non Root User

The installer must be run as a Root user.

Error, <package_name> not installed

Install the package using the following command:

```
# rpm -i software-2.3.4.rpm
```

Error <MEM> is less than the required 8GB memory

Ensure that there is 8 GB of memory available.

Error <TMP> KB is available for TMP, need 100GB

Error <File System Size in KB> KB is available for /opt, need 100GB

The /opt filesystem must have a minimum of 100 GB space for installation.

Error / filesystem requires more than 150 GB of freespace

Ensure that the file system has at least 150 GB available.

Error <Version information> is not supported for IBMPMQ

Uninstall the current DB2 version, and ensure that the system is clean.

Error, Port <portno> is not open

Ensure that the port is open to the firewall, if used.

Error, Connection to <SERVER> on port <PORT> failed

Ensure that the port is open to the firewall, if used.

Unresolved library error on import of PMQDancingCharts or PMQMasterDDLGenerator

PMQDancingCharts and PMQMasterDDLGenerator are applications that are provided with Predictive Maintenance and Quality. When you import these applications into the IBM Integration Toolkit, you get an error.

Symptoms

You see an error message similar to the following example:

```
Project 'PMQDancingChartsJava' is missing required library: 'C:\ProgramData\IBM\
MQSI\shared-classes\junit-4.11.jar'
```

Causes

This issue is caused by references to some unnecessary Java library files in PMQDancingCharts and PMQMasterDDLGenerator application projects.

Diagnosing the problem

Import the project interchange files for PMQDancingCharts and PMQMasterDDLGenerator applications into the IBM Integration Toolkit. Change to the **Problems** view of the IBM Integration Toolkit to see the list of missing library references.

Resolving the problem

To resolve the problem, remove the unnecessary Java library references from the build path of the applications.

1. Change to the Java perspective of the IBM Integration Toolkit.
2. Right-click the **PMQDancingChartsJava** or the **PMQMasterDDLGeneratorJava** project and select **Build Path > Configure Build Path > Java Build Path > Libraries**.
3. The unnecessary Java library references are marked by a red X. Remove these references and click **OK**.
4. Rebuild the project.

SPSS job invocation from the orchestration message flow fails

Operating system or application user credentials have changed, which causes the credentials on IBM Predictive Maintenance and Quality subcomponents to stop working. For example, if the user credentials on the Predictive Maintenance and Quality Analytics node change, you must also change the user credentials that are configured to start SPSS jobs on the Integration Bus node.

Symptoms

The orchestration message flow that calls a job in the event orchestration adapter fails. There is an SPSS credentials error reported in the StdEventLoad_Error and foundation log files.

Resolving the problem

To resolve the problem, you must update the credentials on the Integration Bus node.

1. Log in to the Integration Bus node computer as the **mqm** user.
2. To update both the user ID and password, type the following command:

```
mqsichangeproperties pmqbroker -c UserDefined -o SPSS -n  
UserName,Password -v SPSS_UserID,SPSS_Password
```

Where *SPSS_UserID* is the new user ID and *SPSS_Password* is the new password.
3. To update only the password, type the following command:

```
mqsichangeproperties pmqbroker -c UserDefined -o SPSS -n Password -v  
SPSS_Password
```

Where *SPSS_Password* is the new password.
4. Restart the broker on the Integration Bus node. Type the following commands:

```
mqsisstop broker_name  
mqsisstart broker_name
```

Where *broker_name* is the broker name.

Performance tuning guidelines

You can tune the performance of your IBM Predictive Maintenance and Quality environment.

Deadlock errors happen when parallel processing is enabled

Deadlock errors in IBM Predictive Maintenance and Quality typically happen when parallel processing is enabled by increasing extra instances, and all messages are routed to single folders and queues.

About this task

The error message is named `EventError.txt` and is found in the `\error` folder in the IBM Integration Bus node, location that is defined by the **MQSI_FILENODES_ROOT_DIRECTORY** environment variable.

The error message is as follows:

```
"Error:Label:StdEventLoad_1.LoadEvent:TransactionId:fbcb6b4c0-b434-11e2-8336  
-09762ee50000TransactionTime:2013-05-04 02:34:022322:Child SQL exception:[unixODBC]  
[IBM][CLI Driver][DB2/LINUX8664] SQL0911N The current transaction has been rolled  
back because of a deadlock or timeout. Reason code "2". SQLSTATE=40001"
```

For more information, see “Parallel processing” on page 64.

Procedure

1. Connect to the database with the following command: `db2 connect to db <dbname> [IBMPPMQ]>`
2. Set the isolation level to RR with the following command: `db2 set isolation level to RR`
3. Check the value of the dead lock check time setting with the following command: `db2 get db cfg |grep DL`

The suggested values are:

Interval for checking deadlock (ms)

(DLCHKTIME) = 20000

Deadlock events

(MON_DEADLOCK) = WITHOUT_HIST

4. If the value for the **DLCHKTIME** property is less than 2000, then set the value with the following command: `db2 update db cfg for <dbname> using DLCHKTIME 20000 immediate`
5. Check the value of Lock list and percentage of Locks that are allowed per application `db2 get db cfg |grep LOCK`

The suggested values are:

Maximum storage for lock list (4 KB)

(LOCKLIST) = 100000

Percentage of lock lists per application

(MAXLOCKS) = 97

Lock timeout (sec)

(LOCKTIMEOUT) = -1

Block non logged operations

(BLOCKNONLOGGED) = NO

Lock timeout events

(MON_LOCKTIMEOUT) = NONE

Deadlock events

(MON_DEADLOCK) = WITHOUT_HIST

Lock wait events

(MON_LOCKWAIT) = NONE

6. If the value for the **LOCKLIST** property is less than 1000, then set the value with the following command: `db2 update db cfg for <dbname> using LOCKLIST 100000 immediate`
7. If the value for the **MAXLOCKS** property is less than 97, then set the value with the following command: `db2 update db cfg for <dbname> using MAXLOCKS 97 immediate`

Event processing performance

There are two approaches for increasing the performance of event processing. Events can be processed in multiple threads and events can be processed as a batch.

The event processing flow `StdEventLoad` processes messages that contain a single event or that contain a collection of events. The flow `MultiRowEventLoad` is an example of a flow that loads events and sends them for processing as a collection.

Processing events as collections has the best performance improvement when the events in the collection update the same profile rows. Sort the events so that similar events are processed together. For example, sorting them by device, time, and measurement.

Events that are processed as a collection can be processed only by a single thread. The exception is when the collections that are processed in separate threads do not update any of the same profile rows.

Processing single events by using multiple threads improves performance when the events are updating different profile rows. If the events are all updating the same profile rows, then there is little advantage in using multiple threads. A thread locks the profile rows that it is updating and the other threads must wait until the lock is released. The lock is released when the transaction is committed.

Calculations that are identified as `is_increment` also have improved performance because they can update a profile row in the database without first having to retrieve it and lock it.

Troubleshooting reports

Reports in IBM Predictive Maintenance and Quality are created in IBM Cognos Report Studio. You may encounter problems when using some of the reports included with IBM Predictive Maintenance and Quality.

For further information about troubleshooting reports, see the *IBM Cognos Business Intelligence Troubleshooting Guide*, and the *IBM Cognos Report Studio User Guide*. These documents are available at IBM Cognos Business Intelligence Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSEP7J>).

Audit Report fails with error DMB-ECB-0088 A DMB cube build limit has been exceeded

This error can occur in any report when the master table contains more than 1 million resources but occurs most commonly in the Audit Report.

About this task

To fix the problem, you must increase the **MaxCacheSize** and **MaxNumberOfRecordRows** parameter values in the `qfs_config.xml` file.

Procedure

1. Go to the following IBM Cognos Business Intelligence configuration folder path: `/opt/ibm/cognos/c10_64/configuration`.
2. Open the `qfs_config.xml` file and increase the value of the following parameters:
 - `MaxCacheSize`
 - `MaxNumberOfRecordRows`
3. Save the `qfs_config.xml` file and run the report.

Notices

This information was developed for products and services offered worldwide.

This material may 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. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

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
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Software Group
Attention: Licensing
3755 Riverside Dr.
Ottawa, ON
K1V 1B7
Canada

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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.

The following terms are trademarks or registered trademarks of other companies:

- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.



Index

A

actual values 62
Actual vs plan report 151
Advanced KPI Trend Chart 164
aggregation_type 189, 190
Analytics Solutions Foundation 5
API 23, 175
asset management and manufacturing execution systems
 integration 12
assets 9
Audit Report 159, 237

B

batch processing 62
batch processing events 62
batch_batch 176
benefits 74, 86, 96
business challenges 68, 76, 88

C

calculations 59
calculations, custom 61
carry_forward_indicator 189, 190
city_name 179
Cognos BI artifacts 224
company 28, 38
comparison_string 190
compatibility query mode
 using to see real-time data 214
compatible query mode 213
configure 10
configuring Maximo for OutBound work orders using a web
 service 40
configuring Maximo for OutBound work orders using an XML
 file 42
configuring solution.xml for event flow 64
country_cd 179
country_name 179
creating a workorder 47
custom applications 12

D

dashboard application 4, 12
dashboards 147
 creating 13
data exports Master Data Management 30
data model 215
data_type 190
defect analysis 156
defect summary report 156
Defects by event code 156
Defects by location 156
Defects by production batch 156
defects vs inspection rate line chart 158
DMB-ECB-0088 237
drill through reports 159

E

enabling master data loading in realtime mode 38
environment variables for MDM 27
Equipment control chart 154
Equipment listing report 152
Equipment outliers 155
Equipment profile report 153
Equipment reports 153
Equipment Reports 147
Equipment run chart 155
error messages 233
error reporting 55
event data, configuring 51
event definition 52
event file, sample 217
event format 55
event of type count 59
event processing 51, 62
event type history report 156
event_code 177
event_code_indicator 189
event_code_set 177
event_code_set_name 177
event_type 189

F

feature-based analytics 129
 data preparation 134
 deployment of model 136
 training job 133
Feature-Based Analytics 2
Feature-based predictive model
 input data for training 132
 minimum data requirements 132
 resource sub type level modeling 133
file format 24
file location 24
Fix Central 232
flat file API 175
flat file event input 53
forecast values 62
Framework Manager model database layer 197
Framework Manager model description 197
Framework manager model dimensional layer 213
Framework Manager model logical layer 212
Framework manager model security 213

G

generic batch orchestration 18
group_dim 34
group_type_cd 177
group_type_name 177

H

Health score contributors 148
Health score trend 148

high_value_date 190
high_value_number 190

I

IBM Insights Foundation for Energy 4
IBM Integration Bus 51
IBM Predictive Maintenance and Quality 9
IBM Redbooks 232
import metadata into MDM 31
Incident/recommendation analysis 148
InfoSphere MDM Collaboration Server 23, 28
instrumented assets 9
Integrated analytics
 deployment of model 142
 training job 139
Integrated Analytics 3
Integrated analytics predictive model
 input data for training 138
 orchestration rules 140
Integrated predictive model
 minimum data requirements 138
 resource sub type level modeling 138
Integration analytics
 data preparation 139
interval calculation 59
IS_ACTIVE 175

K

knowledge bases 232
KPI table 55
KPI trending report 151
kpi_indicator 190
KPIs 51, 152

L

language 178
last date of event type 59
Last date of measurement in range 59
last date of measurement type 59
latitude 179
location 34, 179
location example 24
location_name 179
log files 233
longitude 179
low_value_date 190
low_value_number 190

M

Maintenance Advanced Sorting chart 160
maintenance analytics 116, 125
maintenance analytics data 116
Maintenance Health and Failure Detail report 160
Maintenance Overview Report 160
master data 23, 175
Master Data Management 26
master files, sample 217
material usage by process crosstab 159
Material usage by production batch 160
material_cd 180
material_name 180

material_type_cd 180, 190
material_type_name 180
Maximo 34, 39, 146
Maximo Asset Management 12
MDM company archive file 215
MDM guidelines 29
measurement above limit 59
measurement below limit 59
measurement data 51
measurement delta 59
Measurement in range count 59
measurement of type 59
measurement of type count 59
measurement text contains count 59
measurement_type 189
measurement_type_cd 190
message flows 15
metadata 189
model 184
modeling 117, 118
modify a process 24
modify a resource 24

O

operator_cd 184
orchestration 15
 generic batch 18
Outliers 152

P

parallel processing 62
parametric overview 88
parent_process_cd 181
parent_resource_serial_no 184
planned values 62
pre-modeling data 117
predictive maintenance 9
predictive models 115
predictive quality 9
predictive scores 62
predictive scoring 61
Problem Management Reports
 logging 232
 PMR
 See Problem Management Reports
process_cd 181
process_indicator 190
process_kpi 56
process_name 181
process_profile 58
Product quality dashboard 156
product_cd 182
product_name 182
production_batch_cd 176, 182
production_batch_name 182
profile 58
profile calculations 59
profile table 55
profile_calculation 183
profile_calculation_cd 190
profile_indicator 190
profile_units 190
profile_variable 55
profiles 51

Q

- QEWS - Inspection Chart 166
- QEWS algorithm 5
- QEWS parametric 4
- QEWS quality dashboard - inspection 165
- QEWS quality dashboard - inspection detail history 166
- QEWS quality dashboard - parametric 169
- QEWS quality dashboard - parametric detail history 170
- QEWS quality dashboard - warranty 167
- QEWS quality dashboard - warranty detail history 168
- QEWS quality dashboards 165
- QEWS use case
 - parametric 86
- QEWSC parametric chart 170
- QEWSL - Warranty Chart 168
- quality dashboards 5
- quality early warning system 6
- quality inspection overview 68
- query modes 213
- queue 62

R

- real-time data 214
- recommendations 34, 62, 145
- Recommended actions report 153
- region_cd 179
- region_name 179
- related_production_batch_cd 176
- remove events 64
- remove master data 193
- resource 34
- resource example 24
- resource_kpi 56
- resource_name 184
- resource_profile 58
- resource_sub_type 184
- resource_type_cd 184, 185, 190
- resource_type_name 185
- results 74, 86, 96
- rules 145

S

- schema definition for events 55
- scoring 61
- scoring, preventing 146
- sensor health analytics 126
- Sensor Health analytics 121
- Sensor Health predictive model 121
- serial_no 184
- service requests
 - PMR 232
- site overview dashboard 148
- Site Overview Dashboard 147
- software support and RSS feeds 233

- source_system_cd 186
- SPC - Histogram 163
- SPC - X Bar R/S Chart 164
- SPSSTRIGGER 146
- state_province_cd 179
- state_province_name 179
- statistical process control 163
- supplier_cd 187
- supplier_name 187
- supply_cd 180
- Support Portal 231

T

- technical challenges 68, 76, 88
- tenant 187
- threads 62
- Top 10 Contributors dashboard 150
- Top N Failure Analysis Report 160
- TopN Failure Analysis Report 171
- troubleshooting
 - getting fixes 232
 - IBM Redbooks 232
 - identifying problems 231
 - MustGather information 232
 - proven practices documentation 232
 - reports 237
 - software support and RSS feeds 233
 - Support Portal 231
- troubleshooting resources 231

U

- unit_of_measure 189
- updating recommendations 45
- upsert 175
- use case
 - quality inspection 67
 - warranty 74

V

- value_type_cd 188
- value_type_name 188
- video documentation
 - YouTube 232
- viewing recommendations 46

W

- warranty overview 76
- work order creation, disabling 146
- work order service 39
- work orders 34