

IBM Predictive Maintenance and Quality 2.0 Technical Overview

Vrunda Negandhi

Lakshminarayanan Sreenivasan

Randy Giffen

Mohit Sewak

Amaresh Rajasekharan





International Technical Support Organization

IBM Predictive Maintenance and Quality 2.0 Technical Overview

June 2015

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Second Edition (June 2015)

This edition applies to Version 2.0 of the IBM Predictive Maintenance and Quality Solution.

This document was created or updated on June 25, 2015.

© Copyright International Business Machines Corporation 2013, 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
IBM Redbooks promotions	ix
Preface	xi
Authors	xi
Now you can become a published author, too!	xii
Comments welcome	xii
Stay connected to IBM Redbooks	xii
Summary of changes	xv
June 2015, Second Edition	xv
Chapter 1. Introduction	1
1.1 Predictive maintenance and quality concepts	2
1.2 Traditional approaches	2
1.3 Business value	2
1.4 Applying a predictive maintenance and quality solution	3
Chapter 2. IBM Predictive Maintenance and Quality	5
2.1 Solution overview	6
2.1.1 Working with assets	6
2.2 Solution components	7
2.3 Changes in Predictive Maintenance and Quality 2.0	8
2.3.1 Analytics Solution Foundation	8
2.3.2 New method of calculating profiles and KPIs	8
2.4 Deploying the solution	9
2.4.1 Automated installation	9
2.4.2 Deployment topology	10
Chapter 3. Solution architecture	13
3.1 Overview	14
3.2 Installation	15
3.3 Loading master data	16
3.4 Loading events	17
3.5 Storing events	18
3.6 Performing aggregation	19
3.6.1 Scoring	20
3.7 Decision management	21
3.8 Dashboards and delivering recommendations	22
Chapter 4. Master data loading	25
4.1 Overview of master data	26
4.1.1 Elements of master data	26
4.1.2 Flow of master data	27
4.2 Predictive Maintenance and Quality master data management	28
4.2.1 Master data work flows	28
4.2.2 Master data message flow	29
4.2.3 Master data prerequisites	30

4.2.4	Master data file format and structure	31
4.2.5	Master tables and data format	33
4.3	Importing master data using IBM InfoSphere Master Data Management Collaborative Edition	50
4.3.1	Dynamic references	50
4.3.2	Creating a company	50
4.3.3	Importing metadata	51
4.3.4	Configuring the user interface	51
4.3.5	Customizing groups	53
4.3.6	Guidelines for managing master data	54
4.3.7	Configuring data export files	55
4.3.8	Running data exports	56
4.4	Importing master data from IBM Maximo Asset Management	56
4.4.1	Master data mapping	56
4.4.2	Message flow for master data mapping	59
4.4.3	Prerequisites	60
4.4.4	Exporting data from Maximo Asset Management	60
Chapter 5.	Event mapping and processing	61
5.1	Understanding events	62
5.1.1	Event processing	62
5.1.2	Event data format	63
5.1.3	Schema definition of standard event format	66
5.2	Event processing architecture	67
5.2.1	Event loading and mapping	74
5.3	Event processing in batch mode	78
5.4	Event processing configuration parameters	79
5.4.1	Orchestration engine configuration parameters	79
5.4.2	Profile row pool configuration options	80
5.5	Parallel event processing	80
5.6	Custom solution development	81
Chapter 6.	Sensor analytics	83
6.1	Sensor readings and health score analytics	84
6.2	Storing sensor data in PMQ	84
6.3	Understanding sensor data	85
6.4	Preparing sensor data for analysis	85
6.5	Predictive modeling on sensor data	88
6.6	Sensor model deployment and execution	91
6.6.1	Deployment	92
6.6.2	Recommendations	93
6.7	Sensor analytics reports	94
6.7.1	Site Overview Dashboard	95
6.7.2	Maintenance Overview Dashboard	96
Chapter 7.	Maintenance analytics	97
7.1	Data understanding of maintenance analytics	98
7.2	Accessing and processing maintenance data	98
7.2.1	Work order-to-event mapping (batch mode)	99
7.2.2	Work order-to-event mapping (real-time mode)	100
7.2.3	Maintenance event processing	101
7.3	Predictive modeling on maintenance data	102
7.3.1	Maintenance data processing	103
7.3.2	SPSS stream for predicting maintenance schedules	107

7.4	Manipulating maintenance analytics data	108
7.5	Maintenance model deployment and consumption	108
7.5.1	Deployment by using SPSS C&DS Deployment Manager	109
7.5.2	Recommendations from SPSS Analytical Decision Management	109
7.6	Maintenance analytics reports	110
7.6.1	Maintenance Overview Dashboard	110
7.6.2	Maintenance Advance Sorting report	112
Chapter 8.	Integration analytics	113
8.1	Business understanding of integration analytics	114
8.2	Data understanding of integration analytics	114
8.3	Data preparation for integrated analytics	115
8.4	Predictive modeling for integrated analytics	116
8.4.1	Integrated health score model	117
8.4.2	Real-time invocation of the model	117
8.5	Deploying integrated health score models	119
8.5.1	Deployment	119
8.5.2	Recommendations from IBM SPSS Analytical Decision Management	121
8.6	Integrated Analytics Report	123
Chapter 9.	TopN Failure analytics	125
9.1	Business understanding of TopN Failure analytics	126
9.2	Data understanding of TopN Failure analytics	126
9.3	Data preparation for TopN Failure analytics	127
9.4	Predictive modeling for TopN Failure analytics	127
9.5	TopN Failure model invocation	129
9.5.1	TopN Failure Timer	129
9.5.2	TopN Failure Data Preparation	130
9.5.3	TopN Failure Job Integration	130
9.5.4	TopN Failure Analytics Result Processing	131
9.6	TopN Failure model deployment	132
9.7	TopN Failure Analysis Report	133
Chapter 10.	Statistical process control support	135
10.1	Histogram Report	136
10.2	SPC - X Bar R/S Chart	137
10.2.1	X Bar and R charts	137
10.2.2	X Bar and S charts	137
10.2.3	Customizing the SPC - X Bar R/S Chart	138
10.3	Integrating histogram and SPC charts for advanced analysis	139
Chapter 11.	Inspection and warranty analytics	141
11.1	Business scenarios	142
11.1.1	Quality inspections	142
11.1.2	Warranty issues	143
11.2	Inspection analytics	144
11.2.1	Input data	144
11.2.2	Inspection batch data loading message flow	145
11.2.3	PMQ Inspection Timer message flow	146
11.2.4	PMQ inspection real-time data loading flow	146
11.2.5	Inspection Chart	149
11.3	Warranty analytics	151
11.3.1	Data preparation for warranty analytics	153
11.3.2	SPSS Modeler streams	154

11.3.3	SPSS customizable parameters	154
11.3.4	SPSS model invocation	155
11.3.5	Warranty Analytics report	157
Chapter 12.	Integrating IBM Maximo Asset Management	159
12.1	Synchronization of solution data with Maximo	160
12.2	Consuming recommended actions in Maximo	161
Chapter 13.	Additional reports.	163
13.1	Reporting concepts	164
13.2	Predictive Maintenance and Quality reports	164
13.2.1	Accessing reports	165
13.2.2	Site Overview Dashboard	167
13.2.3	Equipment Dashboard	174
13.2.4	Product Quality Dashboard	178
13.2.5	Audit	182
13.2.6	Drill Through Lists	182
13.2.7	Material Usage By Production Batch	186
13.2.8	Advance KPI Trend Chart	187
Appendix A.	Execution, training, and deployment of SPSS models	189
	Model execution	190
	Batch mode	190
	Training mode	190
	Real-time mode	190
	Model Refresh deployment option	191
	Model evaluation	191
	Model training invocation	191
	HSTrainingTimer message flow	192
	TrainingJobIntegration message flow	192
	FileIntegration message flow	192

Notices

This information was developed for products and services offered in the U.S.A.

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

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

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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.

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.

COPYRIGHT LICENSE:

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


Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Cognos®
DataPower®
DB2®
developerWorks®

IBM®
InfoSphere®
Maximo®
Redbooks®

Redpaper™
Redbooks (logo) ®
SPSS®
WebSphere®

The following terms are trademarks of other companies:

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

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

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

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

Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

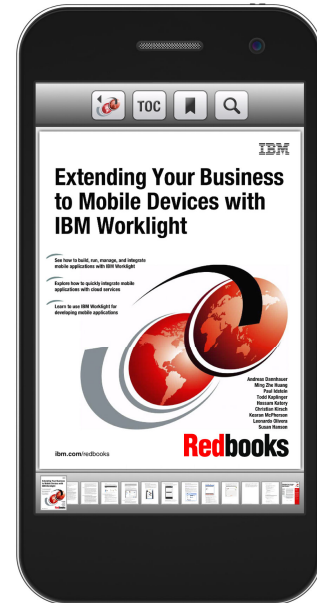
- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

iOS



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redpaper™ publication updated technical overview provides essential details about the data processing steps, message flows, and analytical models that power IBM Predictive Maintenance and Quality (PMQ) Version 2.0.

The new version of PMQ builds on the first one, released in 2013, to help companies efficiently monitor and maintain production assets and improve their overall availability, utilization, and performance. It analyzes various types of data to detect failure patterns and poor quality parts earlier than traditional quality control methods, with the goal of reducing unscheduled asset downtime and improving quality metrics.

Version 2.0 includes an improved method of interacting with the solution's analytic data store using an API from the new Analytics Solution Foundation, a reusable, configurable, and extensible component that supports a number of the solution's analytic functions. The new version also changes the calculation of profiles and KPIs, which is now done using orchestrations that are defined in XML. This updated technical overview provides details about these new orchestration definitions.

Authors

This paper was produced by a team of specialists from around the world working in conjunction with the International Technical Support Organization (ITSO), Raleigh Center:

Vrunda Negandhi is a Certified IT Specialist at IBM in Pune, India. She has more than nine years of experience in the IT industry and currently serves as integration specialist for the Business Analytics Solution team. Vrunda's experience includes work with multiple integration technologies and appliances such as IBM Integration Bus, IBM WebSphere® Message Broker, IBM DataPower®, IBM WebSphere MQ, as well as Java and Java Platform, Enterprise Edition.

Lakshminarayanan Sreenivasan serves as a Senior Certified IT Specialist with IBM in Bangalore, India, with a focus on master data management and business analytics. In his most recent role with IBM, he worked to build analytic solutions that applied IBM SPSS® capabilities to benefit different industry segments. He has contributed to multiple IBM Redbooks® publications and served as speaker at various business analytics technical conferences.

Randy Giffen is an Architect with IBM in Ottawa, Canada, who works on the Business Analytics Solution team. Previously, he was a member of the WebSphere Connectivity and Process Management Tooling team and worked on the Eclipse project. Randy has 16 years of experience with software development at IBM.

Mohit Sewak worked as a Certified Supply Chain Professional and an analytics architect with IBM in Pune, India, where he was a member of the Business Analytics Solution team. Mohit holds several patents, has authored multiple publications, and has spoken at technical forums on analytics. In previous roles, he contributed to planning, implementing, and operating several different predictive, preventive, and breakdown maintenance systems.

Amaresh Rajasekharan is an Executive Architect in the IBM Analytics group based in Somers, NY, and has more than 18 years of experience in IT. He has worked on solutions

involving process control, real-time systems, e-business, and PLM, with responsibilities in product development, solution architecture, technical sales, and consulting. In his current role, Amaresh defines and adapts new industrial sector solutions and reusable assets. He holds IBM Senior Architect and Open Group Distinguished Architect certifications and membership in IEEE and ACM.

Shawn Tooley, a Technical Writer with the ITSO in Raleigh, NC., also contributed to the development of this publication.

Thanks also to these authors of the earlier edition of this paper. which was published in September 2013:

- ▶ Debasish Dash
- ▶ Jagadish Ramakrishnarao

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes made in this edition of the paper and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for IBM Predictive Maintenance and Quality 2.0 Technical Overview
as created or updated on June 25, 2015.

June 2015, Second Edition

This revision includes the following new and changed information.

New information

- ▶ The authors have added new chapters with details about the data processing steps, message flows, and analytical models that enable the advanced functionality of Version 2.0 of the Predictive Maintenance and Quality solution:
 - Sensor analytics
 - Warranty analytics
 - Integration analytics
 - TopN Failure analytics
 - Statistical process controls
 - Inspection and warranty analytics
- ▶ A short appendix has been added with background information and tips for using the component products of the PMQ solution such as IBM SPSS Modeler and related tools.

Changed information

- ▶ Chapters throughout the document have been updated:
 - Additional details about PMQ master data tables and processes
 - Expanded information about event mapping and processing
 - New overviews and screenshots of popular PMQ reports and charts



Introduction

This chapter introduces the concept of predictive maintenance and explains how device monitoring and data-driven analysis can help you anticipate component failures, not just react to them.

This chapter includes the following sections:

- ▶ Predictive maintenance and quality concepts
- ▶ Traditional approaches
- ▶ Business value
- ▶ Applying a predictive maintenance and quality solution

1.1 Predictive maintenance and quality concepts

Predictive maintenance is what its name implies: Maintaining resources and machines, whether large or small, according to *fact-based* expectations for when they will fail or require service. These facts can include the following items:

- ▶ Real-time device status: How is the part performing now?
- ▶ Historical device data: How has the part performed in the past?
- ▶ Data for similar devices: How have other, similar parts performed?
- ▶ Maintenance records: When was the part last serviced or replaced?
- ▶ Maintenance schedules: What does the manufacturer recommendations?
- ▶ Inspection logs: What are the inspection engineers observing during their inspections?
- ▶ Service logs: What are engineers and technicians learning while doing their work?

All of this data is meaningless without analysis. There are hidden patterns lurking within these facts and figures. Decoding these patterns is what powers predictive maintenance and separates it from more traditional, reactionary approaches to equipment repair and replacement.

1.2 Traditional approaches

Predictive maintenance differs considerably from the traditional approaches to determining when to service or replace equipment. For years, companies have kept their production lines running through a combination of these maintenance methods:

- ▶ Reactive or Breakdown: Service or replace equipment after it fails.
- ▶ Preventive: Service or replace equipment according to the manufacturer's suggested schedule, or the amount of time it has been in service, or based on operational observations.
- ▶ Condition Monitoring-based: Service or replace equipment based on monitoring performed to regularly assess its current condition.

The problem with these “old-school” approaches is their high cost. Waiting until a component fails means lost production time and revenue. In-person inspections are expensive and can lead to replacing parts unnecessarily, based only on the inspector's best guess. Following the manufacturer's recommended maintenance schedule saves on inspection costs, but often results in replacing parts that are still functioning well and could continue to do so.

1.3 Business value

A predictive maintenance and quality solution addresses two business use cases:

- ▶ Predict asset failure and extend life: By adopting predictive maintenance and quality, failure of components can be determined based on usage and wear characteristics. In addition, conditions can be identified that lead to high failure rates. This leads to several business benefits:
 - Estimate and extend component life
 - Increase return on assets
 - Optimize maintenance, inventory, and resource schedules

- Predict part quality: A predictive maintenance and quality solution can also detect anomalies within processes, compare parts against their master, and conduct in-depth root cause analysis. This provides the following business benefits:
 - Improve quality and reduce recalls
 - Reduce time to identify issues
 - Improve customer satisfaction

1.4 Applying a predictive maintenance and quality solution

In predictive maintenance, you look for patterns in how equipment is used and the environment in which it is operating, and then correlate this sensor-driven information with any known failures in the equipment. These patterns and correlations are used to create predictive models that are then used to evaluate new sensor data either as it comes in or in batches stored in the analytical warehouse. This process results in predictive scores and forecasts that indicate the relative health of the equipment, the likelihood of future failures, and the best time to perform maintenance on the equipment.

Another aspect of predictive maintenance involves key performance indicators (KPIs) that are collected for each piece of monitored equipment and used primarily 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. These recommendations can be fed into other systems so that users are alerted to them automatically.

If manufacturing defects are on the rise, their causes can often be identified by analyzing data about past operations, environmental conditions, and historical defects. By feeding this information into predictive models, you can predict likely defect rates in the future. 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.



IBM Predictive Maintenance and Quality

This chapter introduces the IBM Predictive Maintenance and Quality solution and the benefits companies can gain by introducing a predictive maintenance system.

This chapter includes the following sections:

- ▶ Solution overview
- ▶ Solution components
- ▶ Changes in Predictive Maintenance and Quality 2.0
- ▶ Deploying the solution

2.1 Solution overview

The IBM Predictive Maintenance and Quality solution helps you monitor, analyze, and report on information gathered from devices and recommend maintenance activities for those devices. With this integrated solution, you can accomplish the following tasks:

- ▶ Predict the failure of a monitored asset so you can fix it and avoid costly downtime.
- ▶ Optimize asset maintenance schedules by incorporating predictive insights and forecasts
- ▶ Learn which equipment parameters (measurement types) best predict failure of a piece of equipment.
- ▶ Conduct statistical process control (SPC) analysis on different operating parameters of an asset.
- ▶ Interactively analyze the frequency distribution histograms based on different operating parameters of an asset.
- ▶ Combine the predictive powers of multiple predictive models, each tapping a variety of equipment and maintenance data in structured, semi-structured, and unstructured formats.
- ▶ Find answers and take corrective actions faster by performing a connected and interactive root cause analysis (RCA) using advanced visualization and predictive analytic techniques and without the need to switch across multiple systems. This process reduces the time to value for identifying the problem, locating its causes and the root cause, and taking corrective action.
- ▶ Provide better early warning signals for accelerated failure rates discovered during inspection of production batches.
- ▶ Provide early warning signals of increasing replacement or wear rates for parts under warranty. Analyze the reasons for the observed rapid replacement or wear, such as changes in service conditions, anomalies during manufacturing, and suboptimal sourcing.
- ▶ Compute customized health score for all equipment using advanced models to predict the patterns in sensor data that can lead to equipment failure.

2.1.1 Working with assets

Monitored assets (also called instrumented assets) generate data such as an asset ID, temperature or other readings, a status code, and a time stamp. This data can be collected and used in analytical models that predict when the asset is likely to fail or require maintenance. Assets that can be monitored for predictive maintenance purposes include equipment used for manufacturing, mining, drilling, farming, and security, covering everything from cars and trucks to engines, cranes, and oil platforms.

Figure 2-1 is a high-level overview of the IBM Predictive Maintenance and Quality solution.

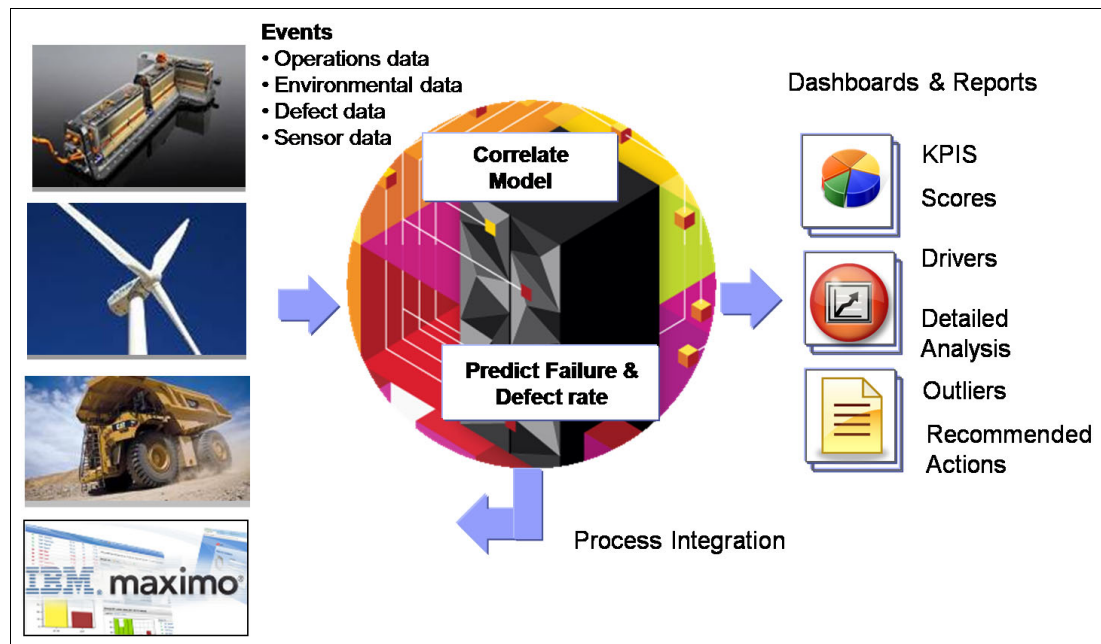


Figure 2-1 High-level overview of the IBM Predictive Maintenance and Quality solution

2.2 Solution components

The IBM Predictive Maintenance and Quality solution has five main components:

- Integration bus layer

The integration bus layer is supported using IBM Integration Bus and connects the internal and external components of the solution. The layer has these main jobs:

- Transforming external events (received from monitored devices) into the format that is required by the solution's database
- Integrating with the solution's predictive-model training and scoring services to obtain the most optimal predictive scores from the most recently trained predictive models. In this way, the recommended maintenance actions are based on the latest received events.

- Data layer

The data layer contains the solution's analytic data store, an IBM DB2® repository for received device events, and any recommended maintenance activities generated in response to those events. The data layer also stores critical master data about devices being monitored. Master data can be loaded into the analytic data store through the integration bus layer, and can be received from an external source or IBM Master Data Management.

- Analytics layer

The analytics layer is supported by IBM SPSS Modeler, IBM SPSS Decision Management, and IBM SPSS Collaboration and Deployment Services. The analytics layer uses models based on historical information and other inputs. These models produce scores that predict the future of an asset and any maintenance recommendations regarding it.

- ▶ Reporting layer

IBM Cognos® Business Intelligence provides a set of live dashboards and reports that show the information captured in the analytic data store. Example reports include a Site Overview report and an Equipment report.

- ▶ Installer

The installer provides automated installation software to configure the software products needed in an IBM Predictive Maintenance and Quality solution. Complex installation steps become easier by using an intuitive user interface.

For more information about the components of the Predictive Maintenance and Quality solution, see Chapter 3, “Solution architecture” on page 13.

2.3 Changes in Predictive Maintenance and Quality 2.0

Predictive Maintenance and Quality 2.0 has the following new features:

- ▶ Analytics Solution Foundation
- ▶ New method of calculating profiles and KPIs

2.3.1 Analytics Solution Foundation

To interact with the analytic data store, Predictive Maintenance and Quality Version 2.0 uses an API from the new Analytics Solution Foundation. This is a reusable, configurable, and extensible component that supports a number of the solution’s analytic functions. The core of the Analytics Solution Foundation is an orchestration engine that processes events and produces recommendations using aggregation, scoring, and decision management techniques.

The Analytics Solution Foundation provides several advantages:

- ▶ An easy, configurable approach to data model customization
- ▶ Less need for custom code to load custom master data
- ▶ A generic Java API to process events (helps meet specific business requirements with minimal changes to the standard solution message flows)
- ▶ A configurable approach to plug-and-play custom SPSS models in the solution
- ▶ Support for multithreading, which allows users to process vast amounts of data concurrently
- ▶ Improved throughput and greater scalability

2.3.2 New method of calculating profiles and KPIs

In the first version of the solution, the calculation of profiles and KPIs was determined by entries in the MAS-TER_PROFILE_VARIABLE table. In version 2.0, profiles and KPIs are calculated according to orchestrations defined in XML.

The MASTER_PROFILE_VARIABLE table is still used, but only to record how the profile and KPI rows were calculated.

2.4 Deploying the solution

The Predictive Maintenance and Quality solution combines multiple IBM software products into a single integrated workbench. This section describes the solution's automated installer and deployment topology.

2.4.1 Automated installation

With so many products and components making up the Predictive Maintenance and Quality solution, an experienced administrator would likely spend several days installing and configuring them all.

Fortunately, IBM provides automated installation software called the Predictive Maintenance and Quality Installer that can do this work with minimal manual intervention. The Predictive Maintenance and Quality Installer has three parts:

- ▶ **Server installer**

The server installer pushes the software stack to each of the configured nodes and triggers the installation process on those nodes. Then, it performs the configuration necessary to enable communication across the nodes and software.

- ▶ **Content installer**

The content installer deploys the Predictive Maintenance and Quality solution content (message flows, predictive models, reports, and so forth) on the respective nodes.

- ▶ **Client installer**

The client installer hosts all the client tools that are required for configuration and modification of the solution content.

2.4.2 Deployment topology

Figure 2-2 depicts the deployment topology of the solution, the required nodes, and the basic configuration of the nodes.

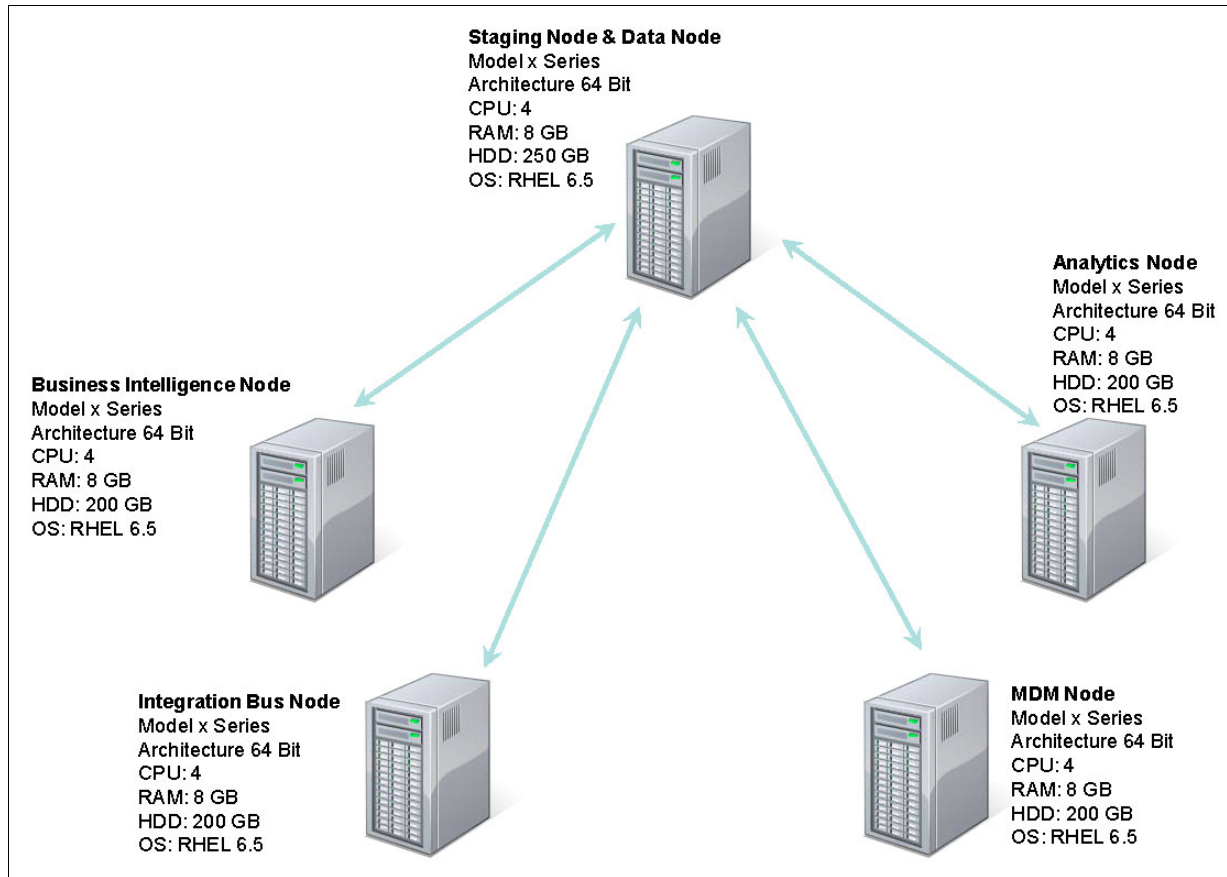


Figure 2-2 Deployment topology

The nodes in the solution serve specific purposes. The following sections provide details about each node.

Predictive Analytics node

The Predictive Analytics node is responsible for generating predictive scores and offering maintenance recommendations based on those scores. The node has the following software installed:

- ▶ IBM WebSphere Network Deployment
- ▶ IBM SPSS Collaboration and Deployment Services
- ▶ IBM SPSS Modeler Server
- ▶ IBM SPSS Data Access Pack
- ▶ IBM SPSS Text Analytics Server
- ▶ IBM SPSS Modeler Collaboration and Deployment Services Adapter
- ▶ IBM SPSS Decision Management
- ▶ IBM SPSS Statistics Server
- ▶ IBM SPSS Statistics Collaboration and Deployment Services Adapter
- ▶ IBM DB2 Client

Business Intelligence node

The Business Intelligence node is where the solution's informative reports and dashboards are produced. It has the following software installed:

- ▶ IBM WebSphere Network Deployment
- ▶ IBM HTTP Server
- ▶ IBM Cognos Business Intelligence
- ▶ IBM DB2 Client

Integration Bus node

The Integration Bus node processes events from monitored devices, updates the content for reports and dashboards, and triggers the predictive scoring and decision management processes. The node also loads master data into the analytic data store. It has the following software installed:

- ▶ IBM WebSphere MQ
- ▶ IBM Integration Bus

Master Data Management node

The Master Data Management node manages the solution's master data, such as the list of monitored devices and their locations and the metadata that is used to determine how event information is processed. This node is optional. You can choose whether to install it. The node has the following software installed:

- ▶ IBM WebSphere Application Server Network Deployment
- ▶ IBM InfoSphere® Master Data Management Collaboration Edition
- ▶ IBM DB2 Client

Data node

The Data node provides the database server that contains the analytic data store. The data store acts as an event store and holds calculated key performance indicators (KPIs) and profiles. It also contains supporting master data for the solution. The node has the following software installed:

- ▶ IBM DB2 Enterprise Server Edition



Solution architecture

This chapter presents the architecture of the IBM Predictive Maintenance and Quality solution. Knowledge of this architecture is key to taking full advantage of the solution's broad configurability and for extending the Predictive Maintenance and Quality solution when custom implementations are required. Understanding the architecture also helps you troubleshoot problems.

This chapter includes the following sections:

- ▶ Overview
- ▶ Installation
- ▶ Loading master data
- ▶ Loading events
- ▶ Storing events
- ▶ Performing aggregation
- ▶ Decision management
- ▶ Dashboards and delivering recommendations

3.1 Overview

The Predictive Maintenance and Quality solution enables the analysis of device events in support of efficient maintenance and product quality. Master data, such as a list of devices and the types of measurements they produce, is loaded into an analytic data store. Device events are supplied to the solution either in real time or as a batch, and are transformed into the format that is required by the solution. The information in the events is recorded in the analytic data store along with aggregated key performance indicators (KPIs) and profiles. The KPIs are accumulated over time and show trends. The profiles indicate the current state of the device and can include statistical calculations of variation. For example, in the example transformer scenario, events that contain the temperature and operating load of the transformer can be aggregated as a KPI of the average temperature and load per day. The operating load can also be aggregated as a profile to record the most recent load and the variability of the load over time.

The analytic data store is implemented as a database by using IBM DB2. The solution provides a script to create the analytic data store from the solution definition XML file that is used to define the data model (master data, events, and profiles). The same XML file can be used to customize the analytic data store to meet specific business needs.

The profile information in the analytic data store and the event data received from monitored devices are used to perform predictive scoring, a process that uses a mathematical model developed in IBM SPSS Modeler to put a numerical value on the likelihood that a device or component failure will occur. These predictive scores are then passed to IBM SPSS Decision Management, which uses a predefined set of rules to determine the appropriate actions to take in response to those scores. For example, if a score indicates that the probability of a transformer failure is less than 0.7 (70%), the rules might call for no immediate action. If the score rises above 0.8 (80%), the rules might trigger a request to have a physical inspection performed. This request can be in the form of a work order created in an external system such as IBM Maximo® Asset Management. The scores and decision management actions are also recorded in the analytic data store as internally generated events, and can be aggregated in the same way as external events from devices.

The information in the analytic data store is viewable in IBM Cognos Business Intelligence (BI) reports. These reports can be used to view the data, such as KPI and profile values, for a particular device over time. These values can also be rolled up to see the combined results for all of the devices at a particular facility. For example, an equipment report can show historical detail about a transformer's operating temperatures or loads. It can also show what recommendations were made, and when, based on the scores returned by a predictive model.

IBM Integration Bus is used to host the orchestration of event processing. Each part of event processing can be viewed as a logically separate step. However, multiple steps can be implemented in a single WebSphere Message Broker flow and more than one step can be implemented in a single compute node within the flow. Events are processed as messages, placed on queues, and then picked up by the next stage in the orchestration.

The simplified architecture for the IBM Predictive Maintenance and Quality solution is shown in Figure 3-1. The following sections in this chapter provide details about customizing event processing for particular scenarios.

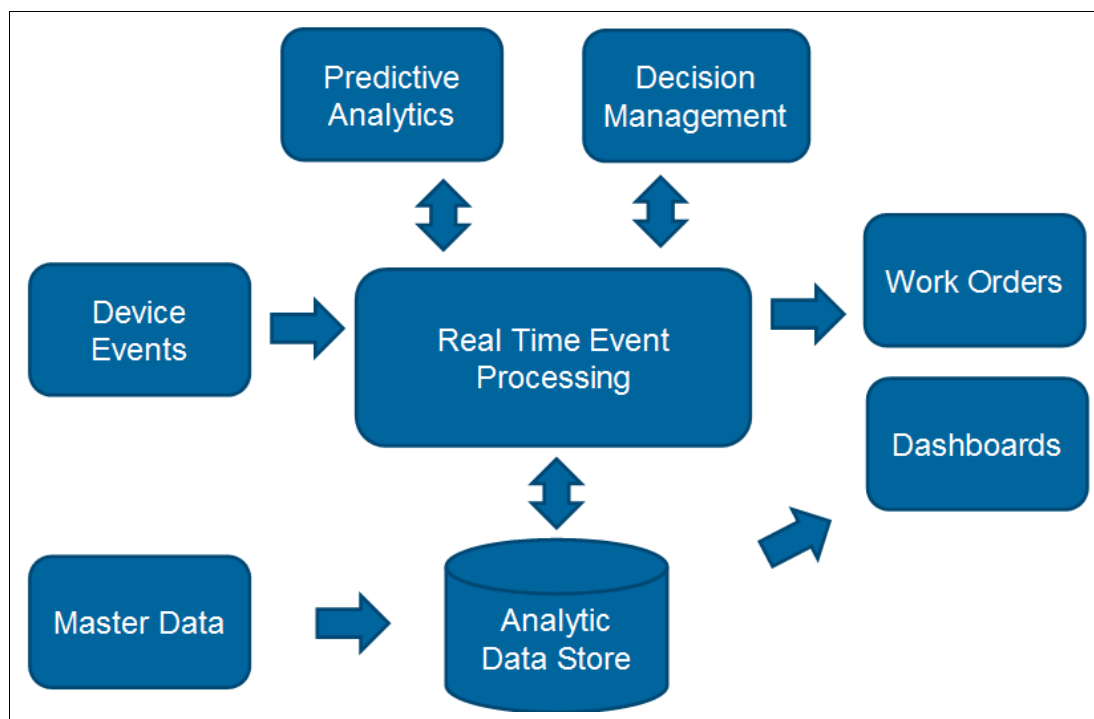


Figure 3-1 Simplified architecture of the IBM Predictive Maintenance and Quality solution

3.2 Installation

An automated installer is provided with the Predictive Maintenance and Quality solution. Installation details are provided in the PMQ 2.0 product documentation available at this location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

The solution can be set up on a cloud platform such as SoftLayer® from IBM. But installing on SoftLayer is different from the typical installation covered in the product documentation. Keep these additional guidelines in mind:

- ▶ Before starting a deployment on SoftLayer, ensure that the hosts files have been updated with the correct names.
- ▶ Make sure to update the `resolv.conf` file with either a SoftLayer DNS entry (a default entry created by SoftLayer) or the DNS entry that you have created on SoftLayer (a custom entry that is created by a user on SoftLayer).
- ▶ In SoftLayer, the deployment model will always be with the firewall set to ON. In addition, the firewall ports must be added using the `iptables` command (`iptables -L`). The custom firewall ports occasionally get flushed out, so the user needs to check the firewall ports (again using the `iptables` command) daily.
- ▶ When installing on SoftLayer, use private or public IP addresses or fully qualified domain names throughout the solution and do not mix and match them. Doing so helps ensure that the installer can run smoothly and there are likely to be no connectivity issues between different target nodes in the installation (ESB, database, and so on).

3.3 Loading master data

Master data must be loaded into the Predictive Maintenance and Quality solution before events can be processed. Master data includes information such as the list of monitored devices (for example, transformers) and their locations. Master data, such as the content of the device events, is recorded in the analytic data store.

The solution includes pre-built Integration Bus flows that load master data from files. Together these flows are known as the flat file API, and the master data files that are supplied to this API must be in the comma-separated value (.csv) file format that is required by the solution. Details of this format are provided in the product documentation.

A code value (or business key) is used to uniquely identify each master data record (device, location, and so on) and is used when referencing a particular record. For example, when a device record must reference a location, it uses the location code. The unique code value is also required because the flat file API is an upsert API, meaning that if a row with the same code exists, it is updated. Otherwise a new row is added.

Master data can be edited using IBM Master Data Management, or you can use a program such as Microsoft Excel to edit it directly. In both cases, the master data is exported as CSV files and loaded into the solution by using the flat file API. Master data can also be imported from external systems such as IBM Maximo Asset Management. But remember that the Integration Bus flows that load data from external systems must include functions to map the data from the external format to the required master data format.

Special master data called metadata is used to help control and record the behavior of the Predictive Maintenance and Quality solution. Details about metadata are described in later sections in this chapter. For now, know that metadata includes measurement types to identify the kinds of observation values present in an event, such as temperature, humidity, or electrical current load. Metadata also includes profile variables that record the calculations performed on each measurement type. In the transformer scenario, an example of a measurement type is the temperature of the transformer, and a profile variable references the method of calculating average temperature value and variation.

Figure 3-2 shows that master data can be loaded into the analytic data store from various sources.

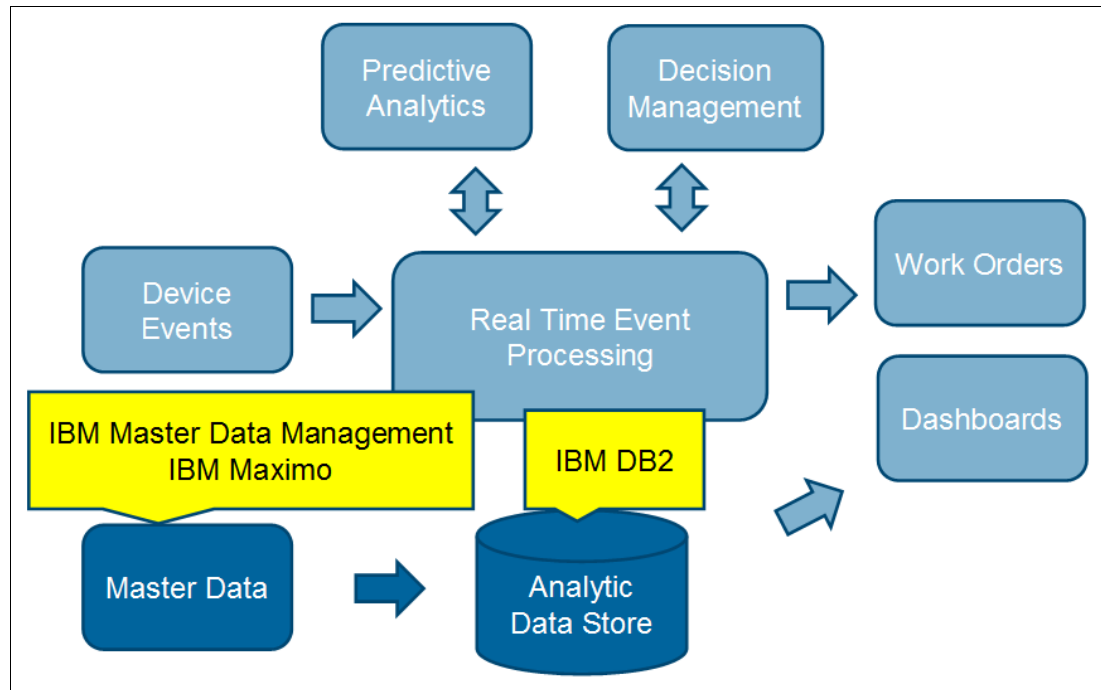


Figure 3-2 Loading master data

3.4 Loading events

Loading events involves delivering them into an Integration bus flow, mapping them to the required event format, and placing them on the PMQ.EVENT.IN queue for processing. The StdEventLoad flow processes the events that are placed on this queue.

Events from a device can be loaded into the Predictive Maintenance and Quality solution in real time using one of the connectivity options provided by Integration Bus, such as a web service or a message queue. A custom message flow is required to transform, or map, the event information from the format used by the transformer monitoring equipment to the format used by the Predictive Maintenance and Quality solution. For example, ILS Technology's deviceWISE is an example of a product that can be used to connect devices to the Predictive Maintenance and Quality solution. A message flow can be created to receive events from ILS deviceWISE using a message queue, map the events to the standard PMQ event format, and then place them on the PMQ.EVENT.IN queue for processing.

Integration Bus offers several options to map event information. Events that can be described using an XML schema (including Data Format Description Language (DFDL)) can be mapped using a graphical mapper. Other events might require the mapping to be implemented in code such as ESQL or Java.

Not all devices can supply events in real time. But these devices can still be supported by loading their events as a batch using a file. The MultiRowEventLoad flow is provided to load a file that contains a batch of events using a predefined format (the DFDL file `multirow_event.xsd` in `PMQEventDataLibrary`). Loading events from a file in a different format involves modifying the DFDL file associated with the file input node in the

MultiRowEventLoad flow. The DFDL file describes the input format of the file and, in this case, supports events that span multiple rows in the file (by using an indicator column).

Figure 3-3 shows that Integration Bus flows deliver real-time event information to the analytic data store.

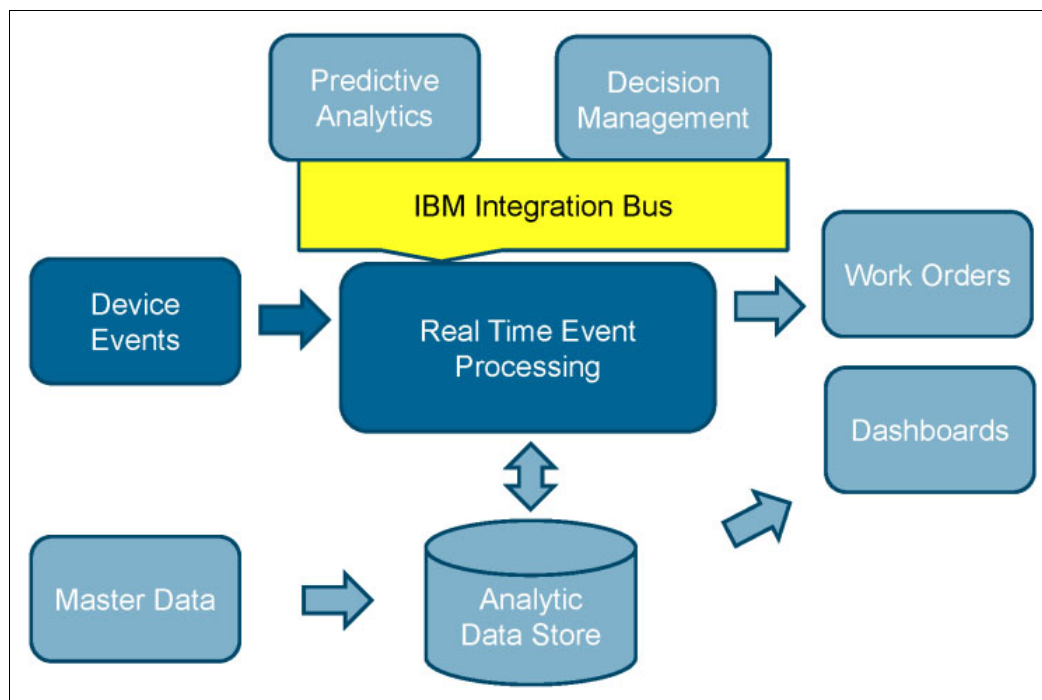


Figure 3-3 Loading events

3.5 Storing events

The previous section describes how device events are delivered to the Predictive Maintenance and Quality solution and mapped to the solution's standard event format. The next step is to store that event information for analysis and processing.

The event processing flow removes events from the PMQ.EVENT.IN queue and stores, or records, them in the solution's IBM DB2 database, called the analytic data store. These events, which can contain many observations, are recorded using the EVENT, EVENT_OBSERVATION, and EVENT_RESOURCE tables. A resource can be a device such as a transformer, or an operator of the device such as the driver of a truck. Events also contain references to master data such as the device location, and this master data must already exist in the analytic data store for the event to be processed. If an event is received from a device that has not yet been loaded into the solution's master data, that event will not be processed. In this case, a message explaining the failure is added to the event processing error log file.

In the example scenario, the transformer supplies events with several observations such as temperature and current load measurements. When these events are loaded, each individual observation is added to the EVENT_OBSERVATION table.

Figure 3-4 shows the products and components that are used to store events.

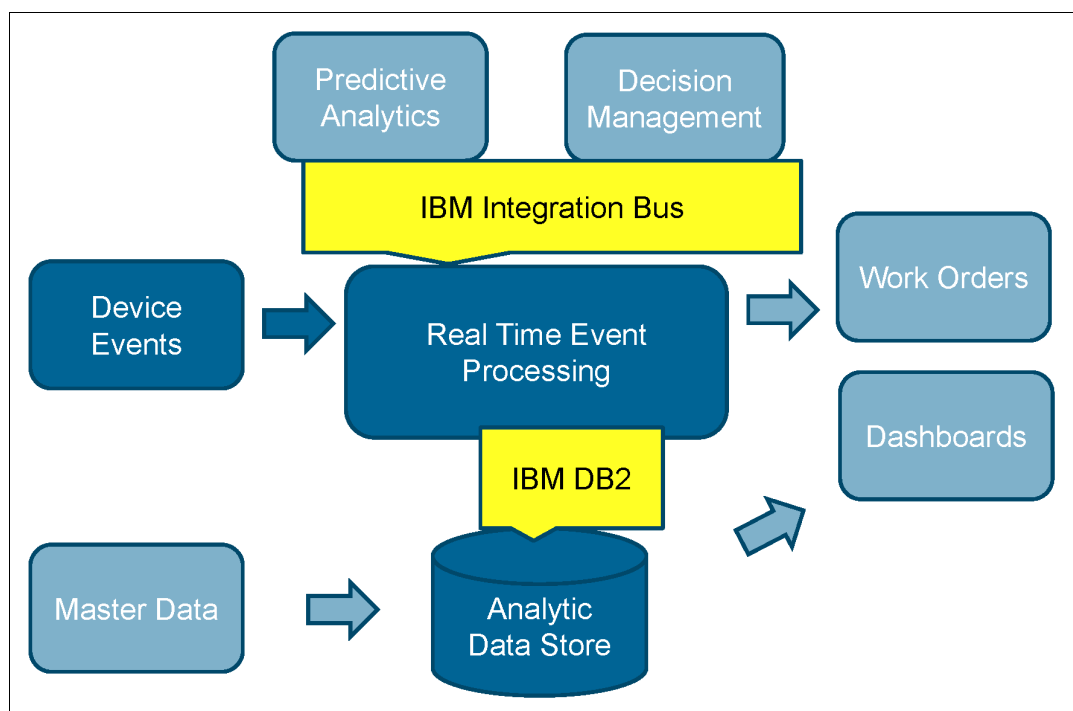


Figure 3-4 Storing events

3.6 Performing aggregation

Events are aggregated into KPIs and profiles. This aggregation is defined as an orchestration (in an orchestration XML file), and is controlled and recorded using measurement type and profile variable master data. An orchestration XML file contains orchestration mappings to define what orchestrations should be performed for a specific event. An orchestration consists of a series of steps. An orchestration step can perform activities such as storing the event, aggregating the event data into KPIs and profiles, and performing scoring and decision management.

The measurement type that arrives with an incoming event defines how the solution must interpret a particular device reading. For example, a reading of 107 can be immediately understood as a temperature and not something else. The measurement types that the solution can process are defined in the master data, and additional measurement types can be added as needed.

Profile variables are also stored in the master data and designate a specific profile calculation that is performed on the incoming data, such as calculating the average and variation of the received values.

During event processing, aggregation is triggered by a combination of event observations (and their accompanying measurement types) and the profile updates that are defined for that measurement type in a profile adapter configuration in the orchestration. The profile update specifies the appropriate calculations for each measurement type. The calculation is what does the work to update rows in the KPI and profile tables during event processing. Each profile variable contains indicators to determine whether a KPI or profile (or both) should be updated.

Several calculations are included with the Predictive Maintenance and Quality solution. Custom calculations can be added to the solution by using the following steps:

- ▶ Load the new calculation using the master data loading flows
- ▶ Define the calculation in the solution XML file
- ▶ Implement the calculation as a Java class and deploy it in the solution

If pursuing these options, see the `PMQ_solution_definition.xml` file in the Integration Bus shared classes path, which is where the supported profile calculations are defined. Additional calculations must be defined there and custom Java implementations need to be developed to support any new calculations.

Figure 3-5 shows the products and components that are used to perform aggregation.

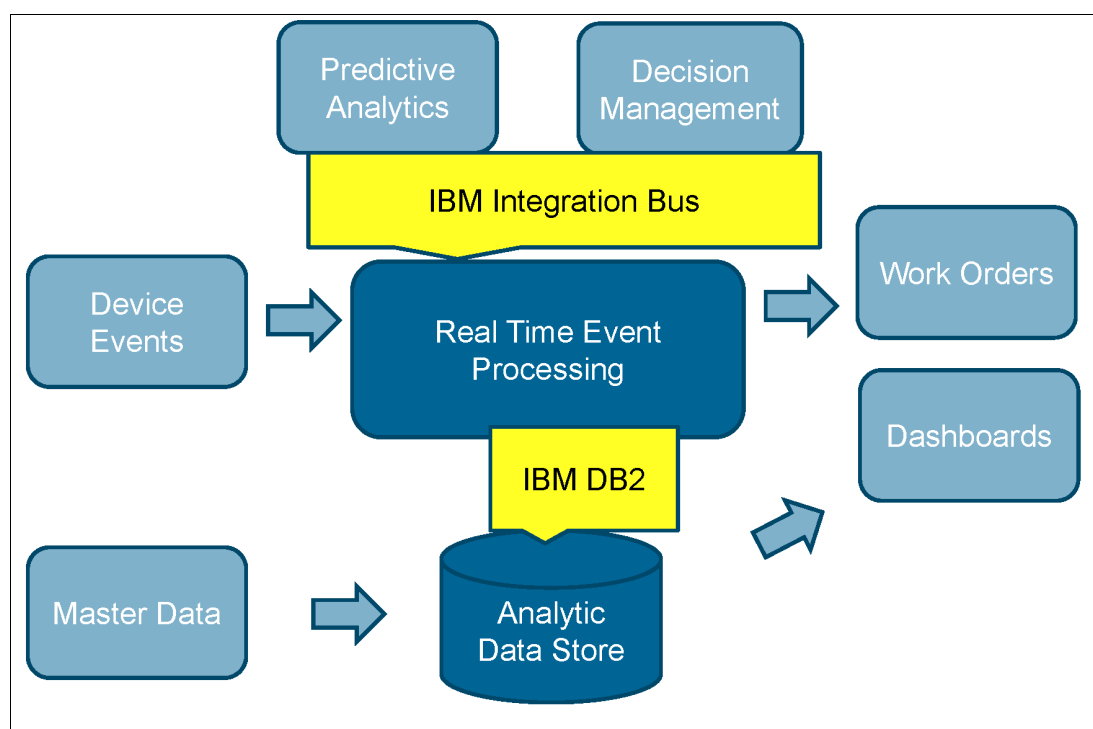


Figure 3-5 Performing aggregation

3.6.1 Scoring

Scoring is a key part of predictive maintenance and involves the use of predictive models that have been trained using historical data to determine the probability of certain future outcomes. For example, a model might be created based on historical data covering transformer temperature, current load, and occurrences of failure.

In the IBM Predictive Maintenance and Quality solution, predictive models are created and trained in SPSS Modeler. The models are then deployed to SPSS Collaboration and Deployment Services, where they are available to be called as scoring web services. These calls are made from a message flow in the `PMQEventLoad` application in Integration Bus, using a service adapter as a step in the orchestration defined for an event.

Each predictive model requires a certain set of inputs. In general, the approach should be to have a profile (a row in a profile table) for each input, but sometimes the input also requires data from a KPI table or the event tables.

The PMQ solution uses APIs provided by Analytics Solution foundation (a reusable, configurable, and extensible component that forms a base to a number of analytic solutions) to prepare the required input data for scoring services and process the scoring service's response. Calling the scoring service in Integration Bus is implemented using a service invocation handler. The results of the scoring service are then recorded and processed in the solution as events.

The profile values for the transformer in the example scenario can include values such as the current temperature and current load, and their recent variation. If a predictive model for transformer failure is created based on these readings, the model can be called as a web service whenever an event is processed. The score that is returned can be thought of as an estimate of the likelihood that the transformer will fail within a designated time, based on the most recent readings.

Figure 3-6 shows the products and components that are used to implement the predictive analytics and decision management components of the solution.

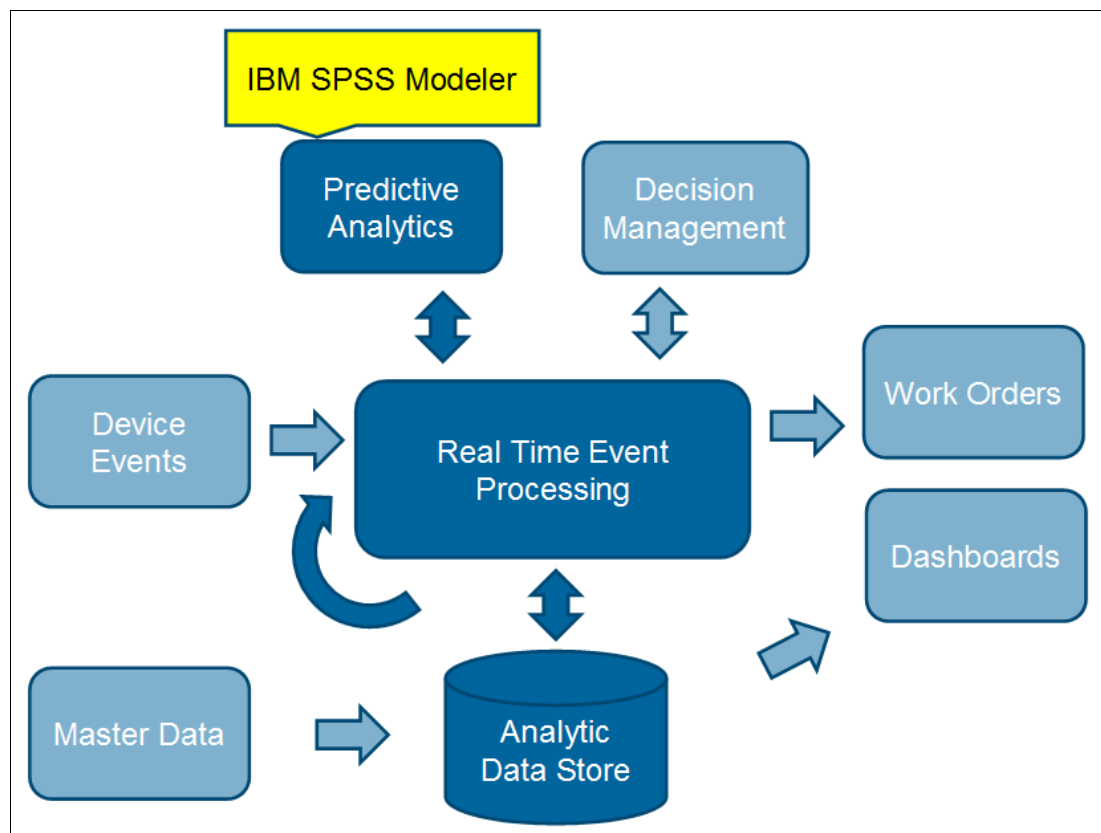


Figure 3-6 Scoring

3.7 Decision management

Decision management allows a predictive maintenance solution to recommend a specific action based on a particular predictive score. This analysis might be performed by using an Integration Bus flow, but the IBM Predictive Maintenance and Quality solution manages the decision-making process and the thresholds for taking action using rules that are developed in IBM SPSS Decision Management. With SPSS Decision Management, rules can be

authored, tested, optimized, and deployed to SPSS Collaboration and Deployment Services, where they become callable as decision management web services.

The decision management web services, as with other web services, require specific input. The solution uses the orchestration defined using Analytics Solution foundation APIs to prepare this input and then receive an action recommendation in reply.

In the transformer scenario, the recommended action that results from the predictive score (which can be based on hours of service, load readings, environmental factors, and so on) might be to perform a detailed onsite inspection to look for early signs of trouble. When the predictive score shows a particularly high probability of failure, the action might be to transfer the load to another device and shut down the transformer for a component-level inspection and possible repair.

As with predictive scores, the same flow that calls the decision management web service records the recommended action by creating an internal event and processing it through the standard event processing flow. As with all events, a specific measurement type and profile variable for the recommendation is required before the profile update can be defined and performed.

Figure 3-7 shows the products and components that are used for decision management.

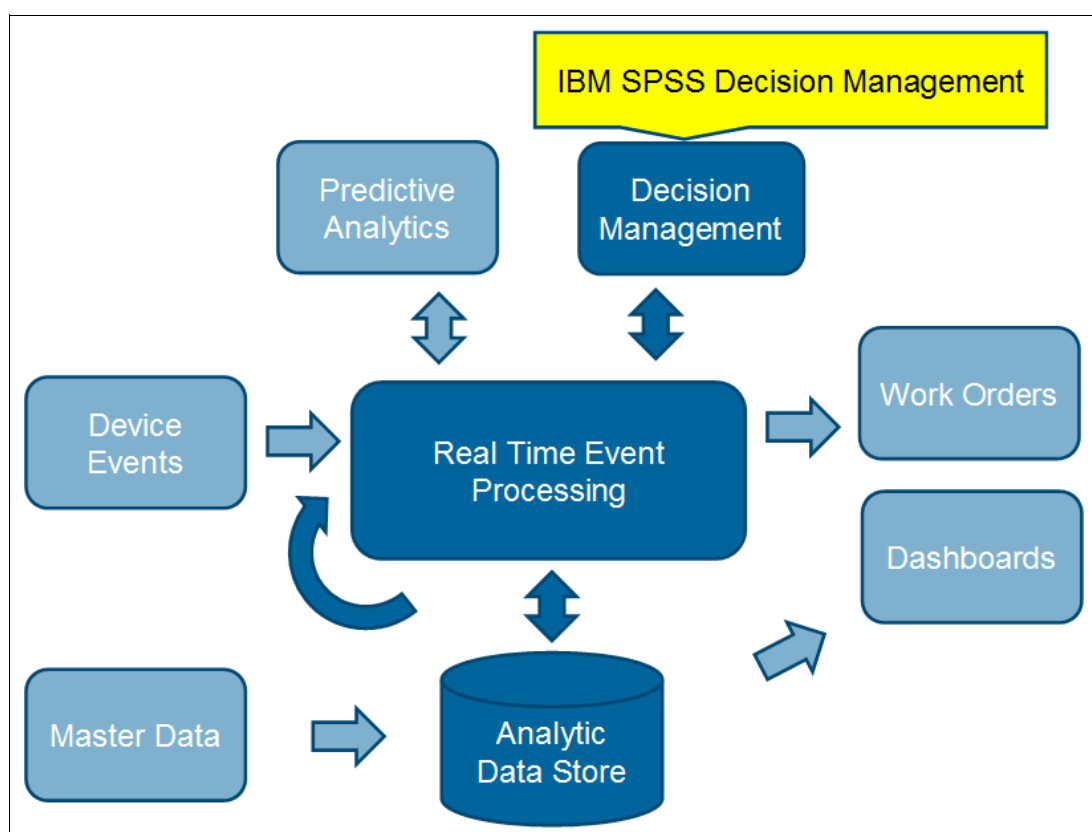


Figure 3-7 Decision management

3.8 Dashboards and delivering recommendations

The ultimate outputs of a predictive maintenance solution are maintenance recommendations for monitored equipment and dashboards that provide views into the health of that equipment.

This part of the solution involves both the communication of recommended actions and the display of aggregated values. With IBM Predictive Maintenance and Quality, the communication of recommended actions can be accomplished with the help of IBM Maximo, while accumulated KPIs and current profile values can be viewed in reports generated by IBM Cognos Business Intelligence.

IBM Maximo is a maintenance application that supports the creation of work orders through a self-generated web service. The Predictive Maintenance and Quality solution includes an Integration Bus flow that can call the Maximo work order web service to create a work order. This flow is triggered when a recommendation for this action is received.

In responding to recommendation for action, a message is placed on the PMQ.MAXIMO.IN message queue by the StdEventLoad message flow. The message is then picked up and processed by the WorkorderCreation message flow. This latter flow is what makes the call to the Maximo work order web service. After the work order is created in Maximo, it becomes visible to authorized individuals. In the transformer scenario, this can include maintenance staff who inspect the transformer or remove the transformer from service based on the specifics of the recommendation. Other work management systems such as IBM Business Process Manager can also be integrated with the Predictive Maintenance and Quality solution in a similar manner. IBM Business Process Manager has a Representational State Transfer (REST) interface for process creation, which can be called based on a recommendation received from the Predictive Maintenance and Quality solution.

The current KPI and profile values calculated by the solution can be viewed in Cognos Business Intelligence reports. These reports are supported by a Cognos Framework Manager Model that accesses the solution's analytic data store. The solution has a set of Cognos Business Intelligence reports including a Site Overview report and an Equipment report.

Figure 3-8 shows the products that are used to implement the work order and components of the solution.

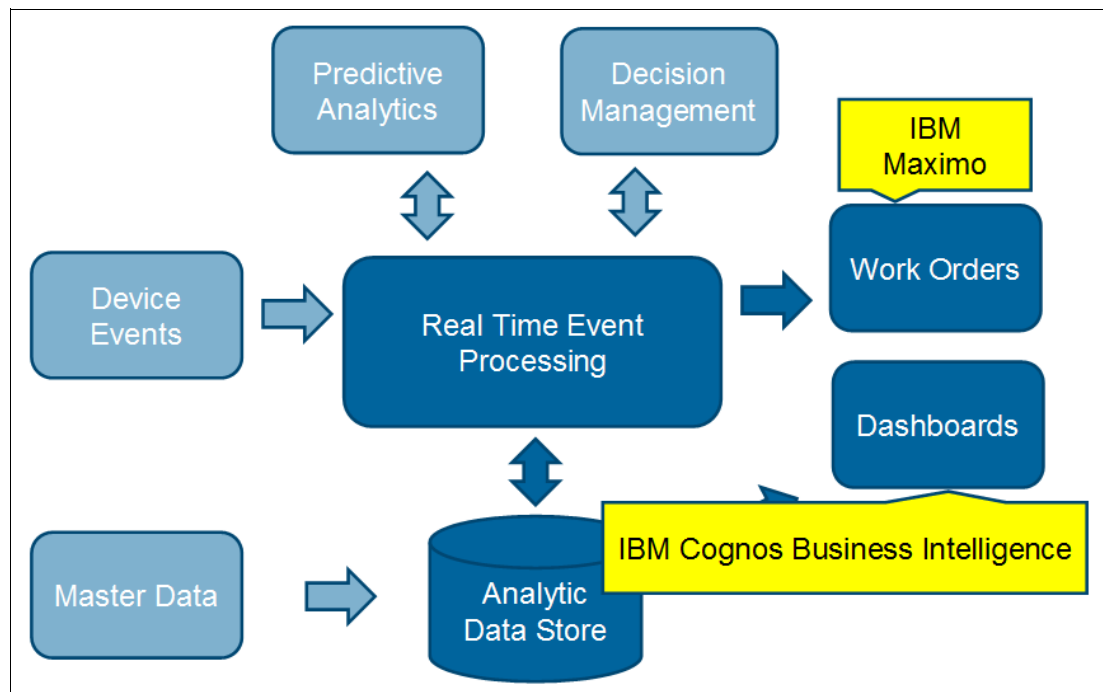


Figure 3-8 Dashboards and delivering recommendations



Master data loading

This chapter provides the process of loading master data into the IBM Predictive Maintenance and Quality solution database, which is known as the analytic data store.

This chapter includes the following sections:

- ▶ Overview of master data
- ▶ Predictive Maintenance and Quality master data management
- ▶ Importing master data using IBM InfoSphere Master Data Management Collaborative Edition
- ▶ Importing master data from IBM Maximo Asset Management

4.1 Overview of master data

Master data is non-transactional data that is updated only when needed, typically based on trigger actions such as the addition of a new piece of monitored equipment, or a change in where the equipment is located. Master data includes information about the sources of transactional data, such as monitored machines (assets) and the humans (agents) that interact with them. It also includes reference data such as the location and manufacturer of each monitored device and the policies or rules that govern what the system does with the transactional data received in device events.

Master data can come from manufacturing engineering systems such as IBM Maximo or from other, existing data sources. IBM InfoSphere Master Data Management can be used to fill in gaps in the data or consolidate data from multiple sources. You can also add attributes, create relationships between items, and define data for which you lack a source. For example, you can classify resources into groups, or add hierarchy information to indicate which pieces of equipment belong to which site or location.

Master data management provides a systematic way of managing the master elements that have the greatest impact on business processes, references, and policies, with version control and without duplication.

The Predictive Maintenance and Quality solution database stores only the elements that are required for predictive analysis. These elements are stored in a solution database called the analytic data store. The solution uses Analytics Solution Foundation APIs to define and load the master data elements. The Foundation APIs use configuration XML files such as a Solution definition file (an XML file that defines the table structure and the relationship between different tables) to generate the master table definition and to load master data.

The solution can be customized by using extra Foundation APIs to add tables for specific customer needs. For more information, see Appendix B of the *Predictive Maintenance and Quality Solution Guide*, which can be downloaded from the PMQ documentation website at:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

4.1.1 Elements of master data

Master data management activities begin during the initial setup of the solution and continue as modifications and additions are made in response to new device events and triggers.

Table 4-1 lists the elements in the solution's master data model.

Table 4-1 Elements of master data

Element	Description
Resource	The entities where events originate. A resource can either belong to a resource type of ASSET (machines or devices) or AGENTS (human operators). Examples include turbines, mining equipment, and operators involved in maintenance or production of resources.
Tenant	A person (individual) or organization (group) using services that are provided by another person or organization. From the Predictive Maintenance and Quality perspective, a tenant might either be an organization or small projects within the organization that use Predictive Maintenance and Quality services for predictive maintenance and product quality for their resources, processes, or materials.

Element	Description
Language	The languages in which information of either master data or event data is stored in the language table. This information is critical to support globalization and ensures that data related to a specific language is grouped and can be queried.
Product	The end output of resources using material in a specified process. Indirectly the parent source of all events will be a product.
Production batch	A set of production units related to the product.
Material	Material used in a production process, such as aluminum used for car engines.
Material type	Defines the material, which can be a broad classification such as artificial or natural.
Supplier	The supplier of the material, which is typically the name of an organization or a person.
Location	The location or place when an event occurs.
Event types	Define the nature of the event, which can be a measurement or a prediction.
Event codes	Link an event with a specific failure observed in the past or occurring for the first time.
Value types	Each event has an associated observation, such as a sensor measurement, and each observation has an associated value type. Value types can be either actual (the event has occurred), planned (the event is scheduled to occur), or forecast (the event is expected to occur in the future).
Measurement type	An event is always associated with an observation, and that observation belongs to a specific measurement type, such as a heat sensor that sends events related to temperature measurement type.
Profile calculations	A set of names provided to identify the various metadata information stored in the MASTER_PROFILE_VARIABLE table supported by the Predictive Maintenance and Quality solution.
Profile variables	Contain the metadata information that is required for processing an event. The metadata information for a resource is a combination of measurement type, event type, process, material, and profile calculations that it supports.

4.1.2 Flow of master data

Master data can come from various external sources. The only requirement is that it adheres to the supported protocols and formats defined in the Master Data chapter of the Predictive Maintenance and Quality Solution Guide. External sources such as IBM Maximo or IBM InfoSphere Master Data Management export the needed information or place it on the designated file system. Integration Bus flows poll these locations and process the files when they arrive. The flows parse the information in the file, check for errors, and then perform an upsert (update or insert) operation on the database.

Figure 4-1 shows the Predictive Maintenance and Quality solution's master data loading process. The master data information from different sources must be present in the CSV format (the formats can vary for each master table and are defined in the Master Data section). The solution's message flows, deployed in Integration Bus, read these files and validate them. Any errors are written to the error log (each master file will have the `_error.txt` suffix) in the `/error` folder. If no errors are observed, the data is committed to the master tables using a JDBC connection.

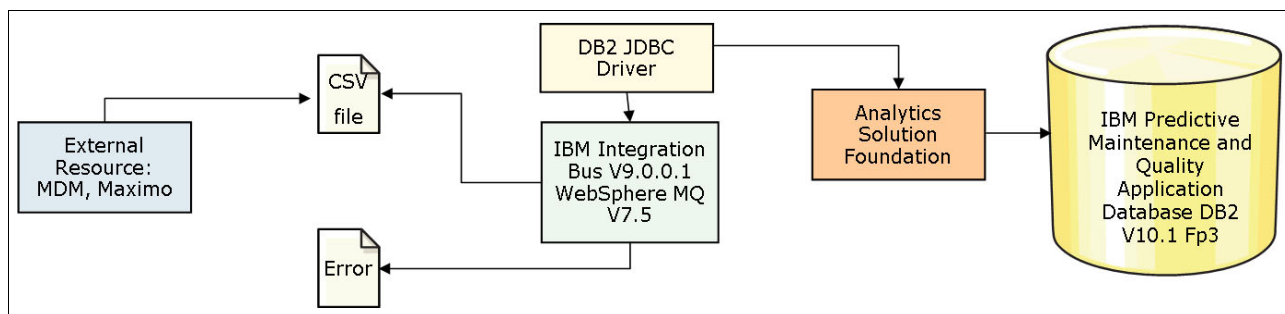


Figure 4-1 Overview of master data work flow

4.2 Predictive Maintenance and Quality master data management

The following sections provide details about the design of the Predictive Maintenance and Quality solution, the prerequisites, and error handling for master data management.

4.2.1 Master data work flows

Master data management in the Predictive Maintenance and Quality solution is handled by Integration Bus flows developed specifically for the purpose. The flows poll defined locations on the file system and process files that meet a predefined naming convention.

The entire database operation that is performed by message flows for either inserting or updating master data in the analytic data store happens in two stages:

1. The files are read and parsed by using the defined schema. If errors are found, a common error management flow is used to write two files to standard error file locations with suffixes added to the original file names to indicate the faulty rows (`_error.csv`) and the actual error information (`_error.txt`). If no errors are found, master data processing proceeds to next stage.
2. Analytics Solution Foundation APIs are started to upsert the master data. If errors are reported due to database connectivity problems or other issues, the errors are logged according to the process in the previous step. The detail exception trace is then logged to the `foundation.log` file under the `/log` directory. Upon successful completion of this step, the original file is stored in an archive with the date and time added as a prefix

These stages are depicted in Figure 4-2.

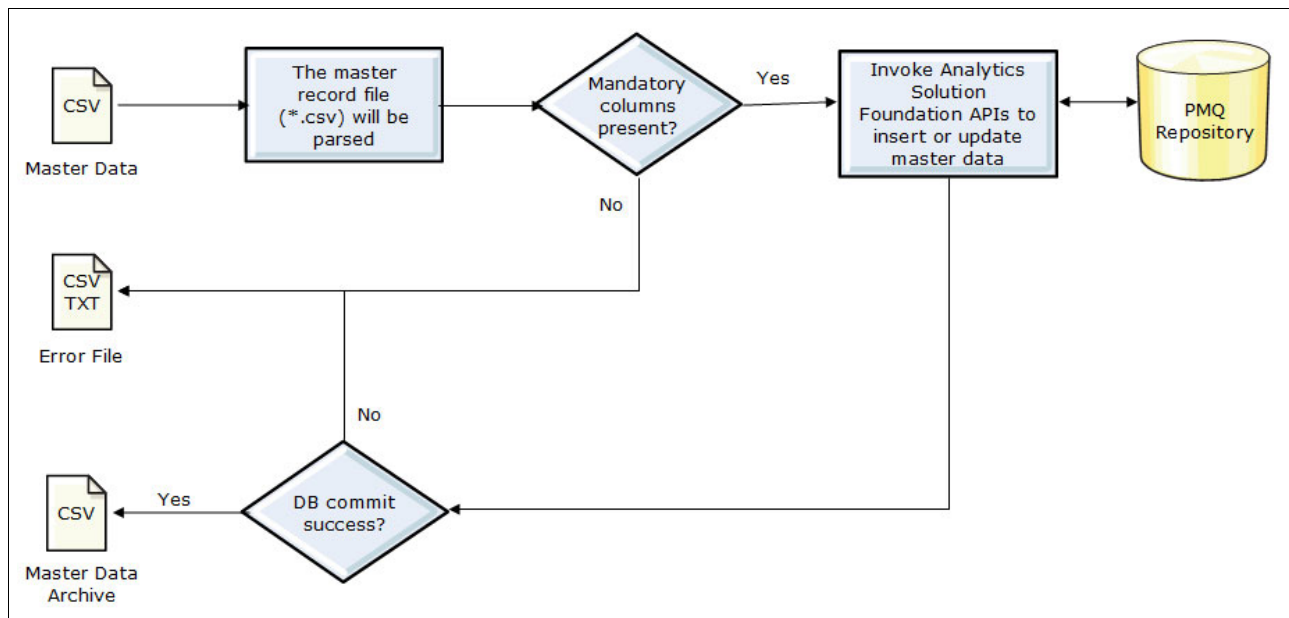


Figure 4-2 Stages of master data work flows

4.2.2 Master data message flow

Integration Bus message flows are used to manage master data work flows. Two flows, in particular, process master data:

- **FileToMQConverter**: This flow converts the master data CSV file into a WebSphere MQ message (see Figure 4-3)

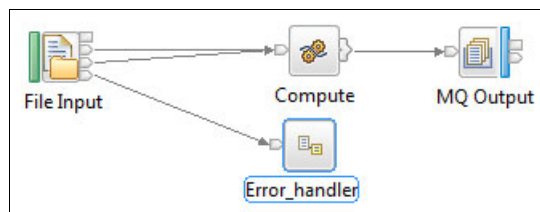


Figure 4-3 FiletoMQ converter message flow

The FileToMQConverter message flow includes these nodes and subflows:

- **File node**: Reads and archives incoming files
- **Compute node**: Extracts the required properties, along with the payload, from the input message and builds an WebSphere MQ output message with MQRFH2 headers
- **MQ Output node**: Puts the WebSphere MQ output message onto the physical message queue
- **Error handler subflow**: Started to write errors to files when errors are detected

- Master data message flow: Reads the WebSphere MQ output message and processes the master data (see Figure 4-4)

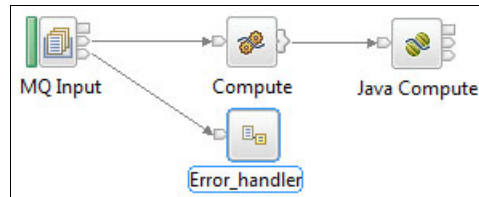


Figure 4-4 Master data message flow

The Master data message flow includes these nodes and subflows:

- MQ Input node: Reads the WebSphere MQ output message provided by the FileToMQConvertor flow
- Compute node: Extracts the message header information to identify the appropriate master data table to be updated, and parses the incoming message using the appropriate Data Format Description Language (DFDL) schema. DFDL schemas define the data structure of a message.
- Java Compute node: Prepares the message data for the appropriate Analytics Solution Foundation API and starts the API to upsert the master data.
- Error handler subflow: Started to write errors to files when errors are detected.

4.2.3 Master data prerequisites

The following prerequisites must be met before the Predictive Maintenance and Quality solution message flows for master data management can be used:

- The required hardware and software are available.
- The server components of the Predictive Maintenance and Quality solution are installed using the Predictive Maintenance and Quality Installer.
- The MQSI_FILENODES_ROOT_DIRECTORY environment variable is set to a proper existing location on the node where Integration Bus is installed. The user who places the CSV files that contain the master data files must be authorized to access this folder.
- Folders named masterdatain, control, and error are present in the location defined by the MQSI_FILENODES_ROOT_DIRECTORY environment variable.
- The solution ESB node's shared-classes folder (/var/mqsi/shared-classes) must contain Analytics Solution Foundation jars and Foundation-related configuration XML files. These artifacts are required by the solution to start the Foundation APIs listed in Chapter 3, "Solution architecture" on page 13.

4.2.4 Master data file format and structure

The master data must be provided in a predefined format (CSV file) and structure (columns). The CSV file format is a comma-separated flat file. The CSV file consists of two sections: The header and the data. The header occupies the first row, and everything else is the data.

The whole file is read by an Integration Bus message flow and parsed row by row. Rows containing incorrect information or where errors are encountered are not processed and are written to error files. Everything else is written to the analytic data store.

To identify the records supplied to the Predictive Maintenance and Quality solution, each record is uniquely identified using a code value (or combination of values). These code values are sometimes called business keys. Because it is a unique identifier for a row in a file, this code is used in other files as a way to reference that particular row. So, for example, in a file that contains a list of resources, the row for a particular resource can contain a location value or code that can be used to identify a location record.

Sometimes a code is required but might not be applicable. For example, in some scenarios clients might not have the dependent code such as a location for a resource because they do not have multiple locations. To support such scenarios, Predictive Maintenance and Quality introduces -NA- (where NA means not applicable). In this scenario, the code is required but might not be used for business purposes.

In addition to a code value, a record typically has a name value. Because both of these values are strings, they might be identical. However, although the code value must be unique for each row and is not normally visible to users, the name is a label that is displayed on reports and dashboards. It is important to note that the name value can be changed but the code value cannot.

Figure 4-5 shows the relationships to other tables for the MASTER_RESOURCE table. The codes that are used to identify a resource record are different from other records because a resource record is identified by both serial number (serial_no in the file) and model number (model in the file), or by operator code (operator_cd column in the file). Reference to other tables is made through the combination of a unique code along with language and tenant codes. For example, in the MASTER_RESOURCE table, the resource type for the resource is obtained by taking the values of resource type code (resource_type_cd), tenant code (tenant_cd), and language code (language_cd). This combination is queried in the MASTER_LOOKUP table to get the resource type ID, which is mapped in the MASTER_RESOURCE table.

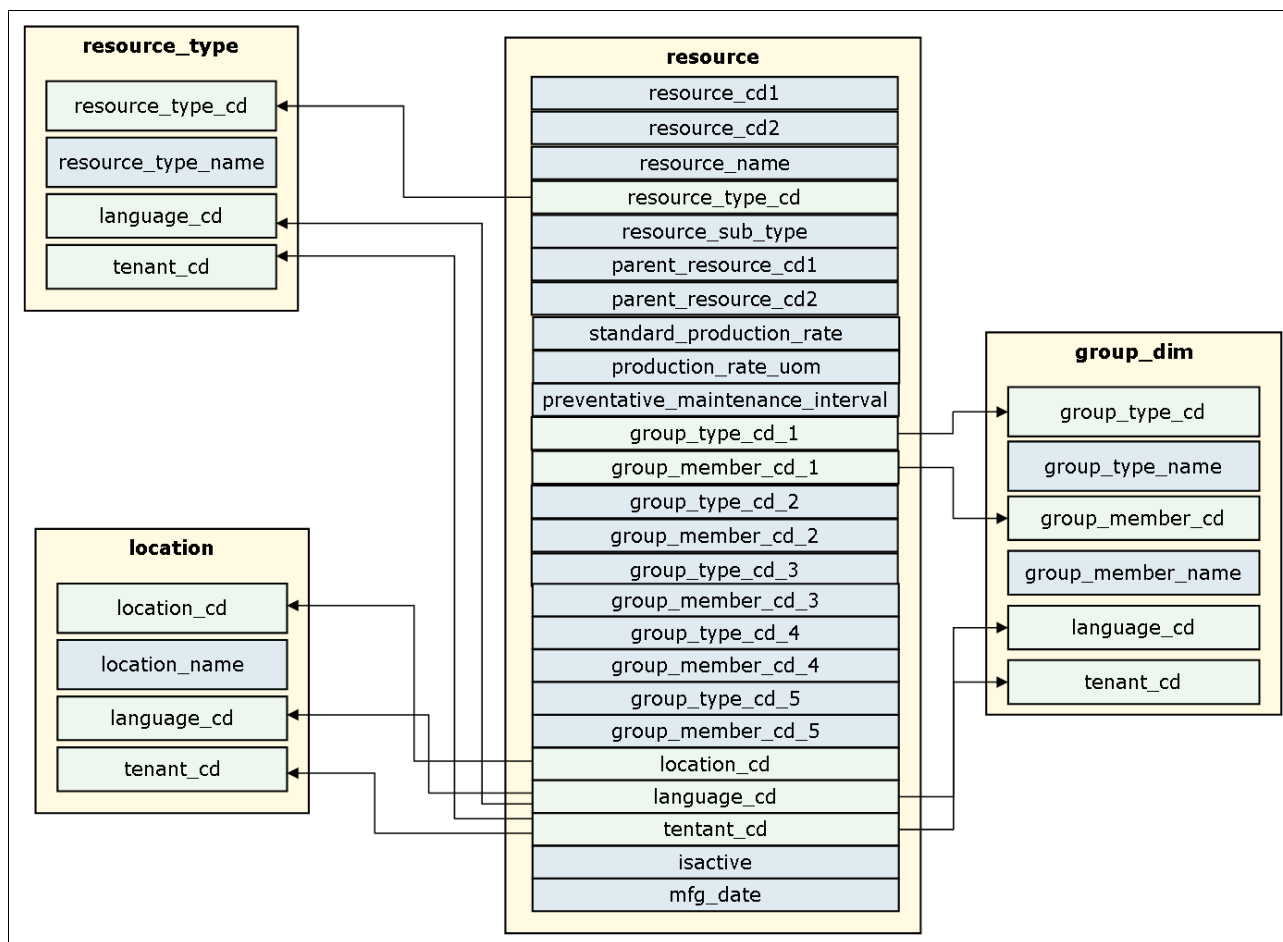


Figure 4-5 Relationship across master information for the MASTER_RESOURCE table

The analytical data store supports hierarchical, parent-child information to be stored for process master entities (up to five levels) and resource master entities (up to 10 levels). When a resource or process parent must be changed, only the modified parent is updated, not the dependent (child) processes or resources. This means that to prevent broken linkages, you must reload the entire resource or process and all related children. You must modify the parent in a master data CSV file that contains all of the appropriate rows and then resubmit the file.

For example, if Resource D had Resource C as a parent and Resource E as a child but must now be changed to show Resource A as a parent, then resource E and its child elements also must be modified to either continue pointing to Resource D or to have Resource C as their new parent.

4.2.5 Master tables and data format

The Predictive Maintenance and Quality solution allows a subset of master data information to be stored in the analytic data store for use in referencing received events and displaying the data on dashboards and in reports. At the time of publication, the solution supports approximately 25 tables that store this important information about resources, processes, materials, and their various dependent elements.

The following sections describe each master table, including the purpose of each table and any dependencies that exist between it and other master tables. Also provided is the file structure for loading the master data, and some sample data for each table.

MASTER_BATCH_BATCH

The BATCH_BATCH table (Table 4-2) creates a many-to-many relationship between production batches. It is intended to be used 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. As an example, suppose Batch 1 splits into 2 and 3, and Batch 3 splits into 4 and 5. In this situation, 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

- Dependency: MASTER_PRODUCTION_BATCH
- File name: batch_batch_upsert*.csv
- Sample (where 1000, 1003, 1004 are all production batches):
production_batch_cd,related_production_batch_cd
1000,1003
1003,1004

Table 4-2 Content of BATCH_BATCH table

Column name	Data type	Required or optional	Remarks
production_batch_cd	String	Required	There is no specific error checking enabled, except that this production batch code exists
RELATED_PRODUCTION_BATCH_CD	String	Required	

MASTER_EVENT_CODE

These records (Table 4-3) contain 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. If the specified event does not exist, it is inserted dynamically during event processing. Typically, the measurement_type of the event defines the event_code_set.

- Dependency: None
- File name: event_code_upsert_*.csv
- Sample:
event_code_set,event_code_set_name,event_code,language_cd,tenant_cd
FAIL,Failure Code X101,X101,EN,PMQ
ALARM,Alarm Code X201,X201,EN,PMQ
ALARM,Alarm Code X202,X202,EN,PMQ
FAIL,Failure Code X203,X203,EN,PMQ
FAIL,Failure Code X204,X204,EN,PMQ

Table 4-3 Content of EVENT_CODE table

Column name	Data type	Required or optional	Remarks
event_code_set	String	Required	Event code name
event_code_set_name	String	Required	Operational name for tracking
event_code	String	Required	Internal code unique and should not be changed for either a new tenant or language
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_GROUP_DIM

Group dimension (Table 4-4) is used for grouping similar resources together, for example, all resources related to drilling assembly or painting assembly or organizations. These records provide classifications for resources. Up to five classifications are possible for each resource. The classifications can vary between instances of the solution.

- Dependency: None
- File name: group_dim_upsert_*.csv
- Sample:
group_type_cd,group_type_name,group_member_cd,group_member_name,language_cd,tenant_cd
ORG,Organization,C1,C1 Department,EN,PMQ
ORG,Organization,C2,C2 Department,EN,PMQ
ORG,Organization,C3,C3 Department,EN,PMQ

Table 4-4 Content of GROUP_DIM table

Column name	Data type	Required or optional	Remarks
group_type_cd	String	Required	Internal code that is unique and should not be changed for the same event in case of either a new tenant or language

Column name	Data type	Required or optional	Remarks
group_type_name	String	Required	Business name
group_member_cd	String	Required	Internal code that is unique and should not be changed for the same event in case of either a new tenant or language
group_member_name	String	Required	Business
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

LANGUAGE

The language record (Table 4-5) is for entering the supported languages and setting the default language.

- ▶ Dependency: None
- ▶ File name: language_upsert_*.csv
- ▶ Sample:


```
language_cd,language_name,isdefault
EN,English,1
JP,Japanese,0
KR,Korean,0
Hi,Hindi,0
Fr,French,0
Ch,Chinese,0
```

Table 4-5 Content of LANGUAGE Table

Column name	Data type	Required or optional	Remarks
language_cd	String	Required	
language_name	String	Required	
isdefault	Integer	Required	Default can be set to 1 or 0. Setting to 1 indicates that the row is the default. Only one row in the table can be marked as the default.

MASTER_LOCATION

Location (Table 4-6) defines places where resources are present and provides location information for the source of events. An example is an engine production room.

- Dependency: None
- File name: location_upsert_*.csv
- Sample:

```
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,Isactive
```

```
Room1,Room1,NA,North America,US,United States,NY,New York,New York City,40.714,-74.0064,EN,PMQ,1
```

```
Room2,Room2,NA,North America,US,United States,MI,Michigan,Detroit,43.686,-85.0102,EN,PMQ,1
```

```
Room3,Room3,NA,South America,US,United States,Peru,Peru,Lima,11.258,-75.1374,EN,PMQ,0
```

Table 4-6 Content of LOCATION table

Column name	Data type	Required or optional	Remarks
location_cd	String	Required	Internal code that is unique and should not be changed for the same event in case of either a new tenant or language
location_name	String	Required	
region_cd	String	Optional	
region_name	String	Optional	
country_code	String	Optional	
country_name	String	Optional	
state_province_cd	String	Optional	
state_province_name	String	Optional	
city_name	String	Optional	
latitude	Double	Optional	Maximum up to 5-digit decimal is supported; more than that results in error
longitude	Double	Optional	Maximum up to 5-digit decimal is supported, more than that results in error
isactive	Integer	Optional	1 or 0; if not provided is treated as active and 1 is passed
tenant_cd	String	Optional	if not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	if not provided, takes default from MASTER_LANGUAGE table

MASTER_MATERIAL

This table (Table 4-7) defines the material that is used for an event. It is defined as a specific instance of a material type, including a link to the supplier. It can represent material used in a repair or material used in a production process.

- Dependency: Material_type and Supplier
- File name: material_upsert_*.csv
- Sample:

```
material_cd,material_name,material_type_cd,supplier_cd,language_cd,tenant_cd,Is  
active  
  
20390,Section 20390,SECTION,WS,EN,PMQ,1  
  
20391,Module 20391,MODULE,WS,EN,PMQ,1
```

Table 4-7 Content of MATERIAL table

Column name	Data type	Required or optional	Remarks
material_cd	String	Required	
material_name	String	Optional	
material_type_cd	String	Required	The code that is provided should exist in the MASTER_MATERIAL_TYPE table; if not present, an error is written
supplier_cd	String	Required	The code that is provided should exist in the MASTER_SUPPLIER table; if not present, an error is written
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table
isactive	Integer	Optional	1 or 0; if not provided, is treated as active and 1 is passed

MASTER_MATERIAL_TYPE

Material Type (Table 4-8) defines the material, whether it is material used in a repair, such as engine filters or other parts, or material used in a production process.

- Dependency: None
- File name: material_type_upsert_*.csv
- Sample:

```
material_type_cd,material_type_name,language_cd,tenant_cd  
PROD,Product,EN,PMQ  
SECTION,Section,EN,PMQ  
MODULE,Module,EN,PMQ
```

Table 4-8 Content of MATERIAL_TYPE table

Column name	Data type	Required or optional	Remarks
material_type_cd	String	Required	Unique member

Column name	Data type	Required or optional	Remarks
material_type_name	String	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_MEASUREMENT_TYPE

Measurement_Type (Table 4-9 on page 39) contains all the measures and event code sets that can be observed for resource, process, and material records. Some examples of measurement types are engine oil pressure, ambient temperature, fuel consumption, conveyor belt speed, and capping pressure. 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 record becomes the event_code_set record, while the measurement_type_name record becomes the event_code_set_name record. This action acts as a trigger to the event integration process to begin recording event codes from the observation_text record.

- Dependency: None
- File name: measurement_type_upsert_*.csv
- Sample:

measurement_type_cd,measurement_type_name,unit_of_measure,carry_forward_indicator,aggregation_type,event_code_indicator,language_cd,tenant_cd

```

SET,Section Test,,0,AVERAGE,1,EN,PMQ
CELLLD,Component Load Test,,0,AVERAGE,1,EN,PMQ
SLT,Section Load Test,,0,AVERAGE,1,EN,PMQ
CLT,Component Life Test,,0,AVERAGE,1,EN,PMQ
ATIME,Assembly Time,hrs,0,SUM,0,EN,PMQ
QTY,Quantity Produced,,0,SUM,0,EN,PMQ
ITIME,Inspection Time,hrs,0,SUM,0,EN,PMQ
RECOMMENDED,Recommended Action,,0,SUM,1,EN,PMQ
FAIL,Failure,,0,SUM,1,EN,PMQ
TEMP,Ambient Temperature,deg C,0,AVERAGE,0,EN,PMQ
RELH,Humidity,%,0,AVERAGE,0,EN,PMQ
REPT,Repair Time,hrs,0,SUM,0,EN,PMQ
OPHR,Operating Hours,hrs,0,SUM,0,EN,PMQ
RPM,RPM,RPM,0,AVERAGE,0,EN,PMQ
INSP,Inspection Count,,0,SUM,0,EN,PMQ
LUBE,Lube Count,,0,SUM,0,EN,PMQ
PRS1,Pressure 1,kPa,0,AVERAGE,0,EN,PMQ
PRS2,Pressure 2,kPa,0,AVERAGE,0,EN,PMQ
PRS3,Pressure 3,kPa,0,AVERAGE,0,EN,PMQ
R_B1,Replace Ball Bearing Count,,0,SUM,0,EN,PMQ
R_F1,Replace Filter Count,,0,SUM,0,EN,PMQ
REPX,Repair Text,,0,SUM,0,EN,PMQ
OPRI,Operating Hours at Inspection,hrs,1,SUM,0,EN,PMQ
REPC,Repair Count,,0,SUM,0,EN,PMQ
HS,Health Score,,0,SUM,0,EN,PMQ
ALARM,Alarm Count,,0,SUM,0,EN,PMQ
LSC,Current Life Span,,0,SUM,0,EN,PMQ

```

Table 4-9 Content of MEASUREMENT_TYPE table

Column name	Data type	Required or optional	Remarks
measurement_type_cd	String	Required	Unique
measurement_type_name	String	Required	Provides a meaningful name
unit_of_measure	String	Optional	Provides information such as RPM, Pascal or N for pressure, C or F for temperature
carry_forward_indicator	Integer	Optional	1 or 0
aggregation	String	Optional	Used mainly by reports to decide on what type of aggregation to perform and display when multiple entries for same coordinates are present, valid values for which include SUM, COUNT, AVERAGE
event_code_indicator	Integer	Optional	1 or 0; 1 indicates that the measurement will have an event_code associated with it, for example FAIL measurement can be used to indicate a particular failure by passing the event_code that is tagged to that specific failure
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_PROCESS

Process (see Table 4-10) is used for production quality reports that identify whether any failures or drawbacks in the process can be detected and require corrective action be taken. For every row in the file, the process_hierarchy is also updated and maintained.

process_hierarchy is supported up to five levels. If a particular process has more than five levels of parent relationships, an error is signaled and no entry is made for the process.

- ▶ Dependency: None
- ▶ File name: process_upsert_*.csv
- ▶ Related table: PROCESS_HIERARCHY
- ▶ Sample:

```
process_cd,process_name,parent_process_cd,language_cd,tenant_cd
SET,Section Test,,EN,PMQ
CELLLD,Component Load Test,,EN,PMQ
SLT,Section Load Test,CELLLD,EN,PMQ
PLT,Product Life Test,,EN,PMQ
CLT,Component Life Test,CELLLD,EN,PMQ
ASSM,Assembly,,EN,PMQ
PA,Product Assembly,ASSM,EN,PMQ
```

Table 4-10 Content of PROCESS table

Column name	Data type	Required or optional	Remarks
process_cd	String	Required	Unique member
process_name	String	Required	

Column name	Data type	Required or optional	Remarks
parent_process_cd	String	Optional	Used for establishing hierarchy relationship
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_PRODUCT

The Product table (Table 4-11) provides the list of products that are covered by the solution.

- ▶ Dependency: None
- ▶ File name: product_upsert_*.csv
- ▶ Sample:


```
product_cd,product_name,product_type_cd,product_type_name,
language_cd,tenant_cd,Isactive
2190890,Product 2190890,B001,Castor,EN,PMQ,1
2190891,Product 2190891,B003,Aix sponsaEN,PMQ,1
2190892,Product 2190892,B004,StrixEN,PMQ,1
```

Table 4-11 Content of PRODUCT table

Column name	Data type	Required or optional	Remarks
product_cd	String	Required	Product code
product_name	String	Required	Product name
product_type_cd	String	Required	Type of product
product_type_name	String	Required	Product type name
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table
isactive	Integer	Optional	1 or 0; if not provided, is treated as active and 1 is passed

MASTER_PRODUCTION_BATCH

Production_Batch (Table 4-12 on page 41) captures all of the products and the batches under which they were manufactured.

- ▶ Dependency: MASTER_PRODUCT
- ▶ File name: production_batch_upsert_*.csv
- ▶ Sample:


```
production_batch_cd,production_batch_name,product_cd,product_type_cd,produce_date,language_cd,tenant_cd
PPR-XXX-001,Castor,PPB-00000004,PPX-00000006,2010-12-01,EN,PMQ
PPB-XXY-003,Melospiza lincolni,PPB-00000004,PPX-00000006,2011-01-01,EN,PMQ
PPC-XXY-005,Procyon lotor,PPB-00000004,PPX-00000006,2011-01-28,EN,PMQ
PPM-XXZ-006,Tagetes tenuifolia,PPB-00000004,PPX-00000006,2011-02-28,EN,PMQ
PPS-XXZ-008,Statice,PPB-00000004,PPX-00000006,2011-04-01,EN,PMQ
PP9-XX9-009,Allium,PPB-00000004,PPX-00000006,2011-07-01,EN,PMQ
```

Table 4-12 Content of PRODUCTION_BATCH table

Column name	Data type	Required or optional	Remarks
production_batch_cd	String	Required	Unique
production_batch_name	String	Required	
product_cd	String	Required	This should exist in MASTER_PRODUCT
product_type_cd	String	Required	This should exist in MASTER_PRODUCT
produced_date	Date	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table
isactive	Integer	Optional	1 or 0; if not provided, is treated as active and 1 is passed

MASTER_PROFILE_CALCULATION

This table (Table 4-13) contains the standard list of profile calculations (business rules) supported by the solution. The file is language-independent and must be modified only when the solution modifies or provides additional profile calculations.

- Dependency: None
- File name: profile_calculation_upsert_*.csv
- List of supported calculations:
 - profile_calculation_name,tenant_id
 - Interval Calculation,PMQ
 - Measurement of Type,PMQ
 - Event of Type Count,PMQ
 - Measurement of Type Count,PMQ
 - Measurement Text Contains Count,PMQ
 - Measurement in Range Count,PMQ
 - Last Date of Event Type,PMQ
 - Last Date of Measurement Type,PMQ
 - Last Date of Measurement in Range,PMQ
 - Measurement Above Limit,PMQ
 - Measurement Below Limit,PMQ
 - Life Span Analysis,PMQ
 - Life Span Analysis Failure,PMQ
 - Delta Calculation,PMQ

Table 4-13 Content of PROFILE_CALCULATOIN table

Column name	Data type	Required or optional	Remarks
profile_calculation_name	String	Required	Standard list
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table

MASTER_PROFILE_VARIABLE

Profile_Variable (Table 4-14 on page 43) contains the metadata information that is required for aggregating and profiling resources, processes, and materials. This is where all computation and alignment is captured for use by the message flows.

- Dependencies: MEASUREMENT_TYPE, PROFILE_CALCUALTION, RESOURCE_TYPE and MATERIAL_TYPE
- File name: profile_calculation_upsert_*.csv
- Sample:

FAIL Above Limit,FAIL Above Limit,Measurement Above Limit,FAIL,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

FAIL Below Limit,FAIL Below Limit,Measurement Below Limit,FAIL,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

INSP Above Limit,INSP Above Limit,Measurement Above Limit,INSP,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

INSP Below Limit,INSP Below Limit,Measurement Below Limit,INSP,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

LUBE Above Limit,LUBE Above Limit,Measurement Above Limit,LUBE,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

LUBE Below Limit,LUBE Below Limit,Measurement Below Limit,LUBE,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

OPHR Above Limit,OPHR Above Limit,Measurement Above Limit,OPHR,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

OPHR Below Limit,OPHR Below Limit,Measurement Below Limit,OPHR,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS1 Above Limit,PRS1 Above Limit,Measurement Above Limit,PRS1,ASSET,,,,,,,,18,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS1 Below Limit,PRS1 Below Limit,Measurement Below Limit,PRS1,ASSET,,,,,,,,7,,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS2 Above Limit,PRS2 Above Limit,Measurement Above Limit,PRS2,ASSET,,,,,,,,15,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS2 Below Limit,PRS2 Below Limit,Measurement Below Limit,PRS2,ASSET,,,,,,,,7,,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS3 Above Limit,PRS3 Above Limit,Measurement Above Limit,PRS3,ASSET,,,,,,,,13,1,1,,AVERAGE,0,0,1,EN,PMQ

PRS3 Below Limit,PRS3 Below Limit,Measurement Below Limit,PRS3,ASSET,,,,,,,,7,,1,1,,AVERAGE,0,0,1,EN,PMQ

RELH Above Limit,RELH Above Limit,Measurement Above Limit,RELH,ASSET,,,,,,,,42,1,1,,AVERAGE,0,0,1,EN,PMQ

RELH Below Limit,RELH Below Limit,Measurement Below Limit,RELH,ASSET,,,,,,,,15,,1,1,,AVERAGE,0,0,1,EN,PMQ

REPT Above Limit,REPT Above Limit,Measurement Above Limit,REPT,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

REPT Below Limit,REPT Below Limit,Measurement Below Limit,REPT,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

REPX Above Limit,REPX Above Limit,Measurement Above Limit,REPX,ASSET,,,,,,,,10,1,1,,AVERAGE,0,0,1,EN,PMQ

REPX Below Limit,REPX Below Limit,Measurement Below Limit,REPX,ASSET,,,,,,,,2,,1,1,,AVERAGE,0,0,1,EN,PMQ

RPM Above Limit,RPM Above Limit,Measurement Above Limit,RPM,ASSET,,,,,,,,1500,1,1,,AVERAGE,0,0,1,EN,PMQ

RPM Below Limit,RPM Below Limit,Measurement Below Limit,RPM,ASSET,,,,,,,,800,,1,1,,AVERAGE,0,0,1,EN,PMQ

Table 4-14 Content of PROFILE_VARIABLE table

Column name	Data type	Required or optional	Remarks
profile_variable_cd	String	Required	Unique to identify profile.
profile_variable_name	String	Required	Business name to display on dashboards.
profile_calculation_name	String	Required	Must exist in MASTER_PROFILE_CALCULATION.
measurement_type_cd	String	Required	Must exist in MASTER_MEASUREMENT_TYPE.
resource_type_cd	String	Required	Must exist in MASTER_RESOURCE_TYPE.
material_type_cd	String	Required	Must exist in MASTER_MATERIAL_TYPE.
profile_units	String	Optional	Describes profile unit of measures, such as RPM, voltage, and hours.
comparison_string	String	Optional	Used internally for many aggregation and profile rules, and should be used when profile calculation such as Measurement Text Contains is used. For more information, see Chapter 5, “Event mapping and processing” on page 61.
low_value	Decimal	Optional	Used for setting the low value; for example, RPM range should be 2000 - 3000 rpm, so set 2000 as low_value and 3000 as high_value.
high_value	Decimal	Optional	
high_value_date	Date	Optional	Last
low_value_date	Date	Optional	
kpi_indicator	Integer	Optional	1 or 0; the 1 indicates aggregation to be performed, and 0 indicates not required. If set to 0, the RESOURCE_KPI and PROCESS_KPI tables will not be updated.

Column name	Data type	Required or optional	Remarks
profile_indicator	Integer	Optional	1 or 0, 1 indicates profiling to be performed and 0 indicates not required. If set to 0, RESOURCE_PROFILE and PROCESS_PROFILE tables will not be updated.
data_type	String	Optional	Indicates what type of data to expect.
aggregation	String	Optional	Used for dashboard for aggregation rule, valid set of values include SUM, AVERAGE.
carry_forward_indicator	Integer	Optional	1 or 0.
process_indicator	Integer	Optional	1 or 0, setting process_indicator to 1 marks process to be aggregated and profiled.
variance_multiplier	Integer	Optional	1 and -1 values. This is mainly used by reports and dashboards for trend results. The 1 indicates that the higher the number the better, and; -1 indicates the lower the number the better.
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table.
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table.

MASTER_RESOURCE

This table (Table 4-15 on page 45) captures the information for all resources (machines and humans), including the subgroup to which they belong and their various behavioral characteristics. The information provided for adding and modifying a resource is used for maintaining resource_hierarchy. The solution supports resource hierarchies up to 10 levels deep.

- ▶ Dependencies: MASTER_GROUP_DIM, MASTER_RESOURCE_TYPE and MASTER_LOCATION
- ▶ File name: resource_upsert_*.csv
- ▶ Related table: RESOURCE_HIERARCHY
- ▶ Sample:

```
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_dim_type_cd_1,group_dim_member_cd_1,group_dim_type_cd_2,group_dim_member_cd_2,group_dim_type_cd_3,group_dim_member_cd_3,group_dim_type_cd_4,group_dim_member_cd_4,group_dim_type_cd_5,group_dim_member_cd_5,location_cd,mfg_date,language_cd,tenant_cd,Isactive
```

```
AAAX1-ZZZZT-TC,XYX,Solar,ASSET,Power Saver,,,,,,,,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,GGR-001,MMN,2010-12-20,EN,PMQ,1
```

```
AAAX2-ZZZZT-TV,XYX,Earth,ASSET,LightWeight,,,,,,,,GGP-002,GGP-002,GGP-002,GGP-002,GGP-002,GGP-002,GGP-002,GGP-002,MMB,2011-01-20,EN,PMQ,1
```

```
AAAX3-ZZZZT-TP,XYX,Lunar,ASSET,Medium Load,,,,,,,,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,GGA-003,MMV,2011-02-18,EN,PMQ,1
```

AAAX4-ZZZZT-TY,YTT,Volcano,ASSET,Power
Saver,,,,,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,GGB-004,MMC,2011-03-20,EN,PMQ,1

AAAY5-ZZZZT-TT,XYX,Aura,ASSET,LightWeight,,,,,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,GGC-005,MMX,2011-04-20,EN,PMQ,1

Table 4-15 Content of RESOURCE table

Column name	Data type	Required or optional	Remarks
resource_cd1	String	Optional	Unique. The combination should be one of these: <ul style="list-style-type: none"> ▶ serial_no and model ▶ operator_cd ▶ serial_no ▶ model and operator_cd
resource_cd2	String	Optional	
resource_name	String	Required	Name of the resource
resource_type_cd	String	Required	Should exist in MASTER_RESOURCE_TYPE
resource_sub_type	String	Optional	Used for grouping the similar resources such as paint shop or assembly
parent_resource_cd1	String	Optional	If provided, the combination can be one of these: <ul style="list-style-type: none"> ▶ serial_no and model ▶ or operator_cd ▶ serial_no ▶ model and operator_cd
parent_resource_cd2	String	Optional	
standard_production_rate	Decimal	Optional	
standard_production_rate_uom	String	Optional	
preventive_maintenance_interval	Integer	Optional	
group_type_cd_1	String	Optional	Required from PMQ perspective; if null is provided, it is replaced with -NA- (which will be populated when installed).
group_member_cd_1	String	Optional	
group_type_cd_2	String	Optional	
group_member_cd_2	String	Optional	
group_type_cd_3	String	Optional	
group_member_cd_3	String	Optional	
group_dim_type_cd_4	String	Optional	
group_type_cd_4	String	Optional	
group_type_cd_5	String	Optional	
group_member_cd_5	String	Optional	
location_cd	String	Required	Should exist in the MASTER_LOCATION table

Column name	Data type	Required or optional	Remarks
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table.
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table.
isactive	Integer	Optional	1 or 0; if not provided, it is treated as active and 1 is passed.
mfg_date	Date	Optional	Manufacturing date of the product.

MASTER_RESOURCE_TYPE

The Resource_Type table (Table 4-16) contains information about resource classifications such as ASSET (machines) and AGENT (humans).

- ▶ Dependency: None
- ▶ File name: resource_type_upsert_*.csv
- ▶ Sample:
 - resource_type_cd,resource_type_name,language_cd,tenant_cd
 - ASSET,Asset,EN,PMQ
 - AGENT,Agent,EN,PMQ

Table 4-16 Content of RESOURCE_TYPE table

Column name	Data type	Required or optional	Remarks
resource_type_cd	String	Required	Unique
resource_type_name	String	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_SOURCE_SYSTEM

The Source_System table (Table 4-17) captures the system that is the source of events, such as IBM Maximo. The internal source system, PREDMAIT, is required because it is used for identifying events that have been scored for predictive maintenance.

- ▶ Dependency: None
- ▶ File name: source_system_upsert_*.csv
- ▶ Sample:
source_system_cd,source_system_name,language_cd,tenant_cd,Isactive
PREDMAIT,Predictive Maintenance,EN,PMQ,1
MDM,MDM,EN,PMQ,1
MAXIMO,Maximo,EN,PMQ,1
SENSOR,SENSOR,EN,PMQ,1

Table 4-17 Content of SOURCE_SYSTEM table

Column name	Data type	Required or optional	Remarks
source_system_cd	String	Required	Unique
source_system_name	String	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table.
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table.
isactive	Integer	Optional	1 or 0; if not provided is treated as active and 1 is passed.

MASTER_SUPPLIER

The Supplier table (Table 4-18) contains details of the supplier of the material.

- ▶ Dependency: None
- ▶ File name: supplier_upsert_*.csv
- ▶ Sample
supplier_cd,supplier_name,language_cd,tenant_cd,Isactive
WS,Widget Part Supplier,EN,PMQ,1

Table 4-18 Content of SUPPLIER table

Column name	Data type	Required or optional	Remarks
supplier_cd	String	Required	Unique
supplier_name	String	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table
isactive	Integer	Optional	1 or 0; if not provided is treated as active and 1 is passed

TENANT

Tenant (Table 4-19) contains the records of projects or organizations that want to use the solution independently, but with a common repository.

- ▶ Dependency: None
- ▶ File name: tenant_upsert_*.csv
- ▶ Sample
tenant_cd,tenant_name,isdefault
PMQ,PMQ,1

Table 4-19 Content of TENANT table

Column name	Data type	Required or optional	Remarks
tenant_cd	String	Required	
tenant_name	String	Required	
isdefault	Integer	Optional	1 or 0, if not provided it is treated as passive and 1 is passed for setting it as default

MASTER_VALUE_TYPE

Value Type (Table 4-20) captures how the event occurred and typically falls into one of three types:

- ▶ ACTUAL: The event that occurred
- ▶ PLAN: What was expected to happen
- ▶ FORECAST: A prediction that a particular event will occur

Dependency, file name, and a sample are shown here:

- ▶ Dependency: None
- ▶ File name: value_type_upsert_*.csv
- ▶ Sample:
value_type_cd,value_type_name,language_cd,tenant_cd
ACTUAL,Actual,EN,PMQ
PLAN,Plan,EN,PMQ
FORECAST,Forecast,EN,PMQ

Table 4-20 Content of VALUE_TYPE table

Column name	Data type	Required or optional	Remarks
value_type_cd	String	Required	Unique member
value_type_name	String	Required	
tenant_cd	String	Optional	If not provided, takes default from MASTER_TENANT table
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

MASTER_OBSERVATION_LOOKUP

Observation_Lookup (Table 4-21) captures observation code and observation text mappings.

- Dependency: None
- File name: observation_lookup_upsert_*.csv
- Sample:
observation_cd,observation_cd,language_cd
1001,Consider Postponing the maintenance,EN
1002,Consider Postponing the maintenance,EN
1003,Consider Postponing the maintenance,EN
Table 4-21 Content of OBSERVATION_LOOKUP table

Table 4-21 Content of OBSERVATION_LOOKUP table

Column name	Data type	Required or optional	Remarks
observation_cd	String	Required	Unique member
observation_text	String	Required	Observation text
language_cd	String	Required	If not provided, takes default from MASTER_LANGUAGE table

MASTER_RESOURCE_PRODUCTION_BATCH

Resource Production Batch (Table 4-22) captures the mapping between the resource commissioned in the MASTER_RESOURCE table and the production batch registered in the MASTER_PRODUCTION_BATCH table.

- Dependency: MASTER_RESOURCE, MASTER_PRODUCTION_BATCH
- File name: resource_production_batch_upsert_*.csv
- Sample:
SERIAL_NO,MODEL_NO,PRODUCTION_BATCH_CD,QTY,LANGUAGE_CD

AAAX1-ZZZZT-TC,YXY,PPR-XXX-001,10,EN

AAAX2-ZZZZT-TV,XYY,PPB-XXY-003,10,EN

AAAX3-ZZZZT-TP,YXY,PPC-XXY-005,10,EN

Table 4-22 Content of MASTER_RESOURCE_PRODUCTION_BATCH table

Column name	Data type	Required or optional	Remarks
observation_cd	String	Required	Unique member
observation_text	String	Required	Observation text
language_cd	String	Optional	If not provided, takes default from MASTER_LANGUAGE table

4.3 Importing master data using IBM InfoSphere Master Data Management Collaborative Edition

This section guides you through the postinstallation process of configuring IBM InfoSphere Master Data Management Collaborative Edition to work with the IBM Predictive Maintenance and Quality solution. These steps assume that installation has already been completed. It is also assumed that the software is running on Red Hat Enterprise Linux 6.5.

For more information about these topics, see the IBM Predictive Maintenance and Quality 2.0.0 documentation available at this location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

4.3.1 Dynamic references

IBM Master Data Management Collaboration Server tasks use several dynamic references. Table 4-23 describes these variables.

Table 4-23 Dynamic references

Reference	Description
\$PMQ_HOME	The home directory of the IBM Predictive Maintenance and Quality installation.
<mdmce_install_dir>	The root directory of the InfoSphere Master Data Management Collaborative Edition installation.
<mdm_server_ip>	The IP address of the InfoSphere Master Data Management Collaborative Edition server.
<pmq_mdm_content_zip>	The full path to the content compressed file on the server file system.
<mdm_data_export_dir>	The directory, mount point, or symbolic link on the InfoSphere Master Data Management Collaborative Edition server where data exports are configured to be written. The default is the following directory: \$PMQ_HOME/data/export/mdm.
<wmb_server_ip>	The IP address of the WebSphere Message Broker server.
<wmb_fileapi_input_dir>	The local or remote directory where the Predictive Maintenance and Quality Flat File API expects input data files to be placed for loading into the analytic data store
<company_code>	The company code for InfoSphere Master Data Management Collaborative Edition.
<company_name>	The display name of the company.

4.3.2 Creating a company

Before the solution content can be imported into InfoSphere Master Data Management Collaborative Edition, a company must be created. A company is similar in concept to what other tools refer to as a project. Use the following steps:

1. Stop the InfoSphere Master Data Management Collaborative Edition service:
 - a. Change to the <mdmce_install_dir>/bin/go directory:

```
cd <mdmce_install_dir>/bin/go
```
 - b. Run the **stop_local.sh** command:

```
/stop_local.sh
```


2. Run the Company creation script:
 - a. Change to the `<mdmce_install_dir>/bin/db` directory:


```
cd <mdmce_install_dir>/bin/db
```
 - b. Run the **create_cmp.sh** command:


```
./create_cmp.sh -code=<company_code> --name=<company_name>
```
3. Start the InfoSphere Master Data Management Collaborative Edition service:
 - a. Change to the `<mdmce_install_dir>/bin/go` directory:


```
cd <mdmce_install_dir>/bin/go
```
 - b. Run the **stop_local.sh** command:


```
./start_local.sh
```

After the company is created, you must log in and verify it. Table 4-24 shows the default users that can be used to log in the first time. The leading practice is to then change each default password to something more secure.

Table 4-24 Default users for a newly created company

Role	User name	Password
Administrator	Admin	trinitron
Basic User	Basic	trinitron

4.3.3 Importing metadata

The Predictive Maintenance and Quality metadata content must be imported into InfoSphere Master Data Management Collaborative Edition before it can be used to manage event data. The following command string is used to import the metadata (the arguments are shown on separate lines, but can be entered on a single line):

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

In practice, the string might resemble the following example:

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

The `$TOP` variable is a built-in InfoSphere Master Data Management Collaborative Edition environment variable that points to the root InfoSphere Master Data Management Collaborative Edition directory.

4.3.4 Configuring the user interface

To more easily manage data, configure the InfoSphere Master Data Management Collaborative Edition user interface to expose the Predictive Maintenance and Quality objects in the InfoSphere Master Data Management Collaborative Edition navigation area.

To add objects to the navigation area, select the objects from the Modules menu and click the plus sign (+), as shown in Figure 4-6.

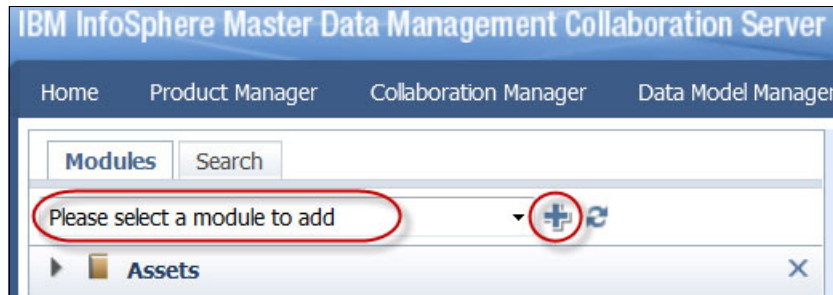


Figure 4-6 Adding a module

The list of available modules is shown in Figure 4-7.

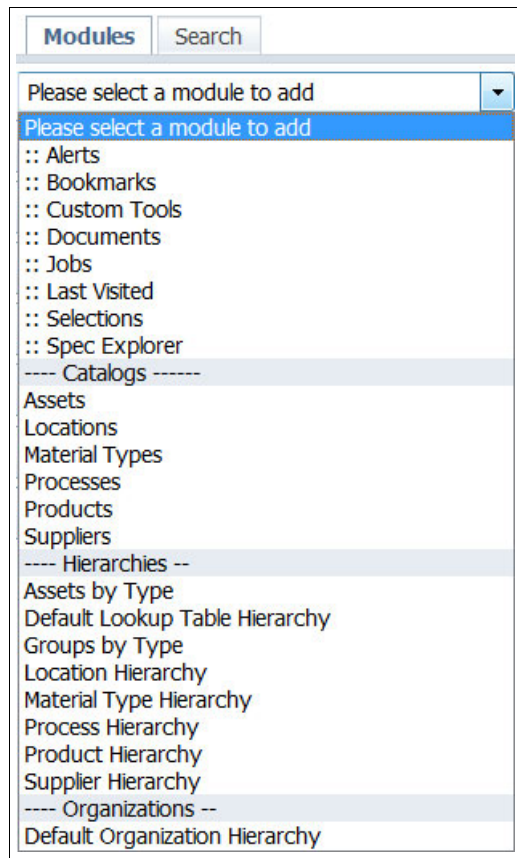


Figure 4-7 List of modules

Table 4-25 lists modules that are typically included in the InfoSphere Master Data Management Collaborative Edition navigation area.

Table 4-25 Typically used modules

Module type	Module
Catalog	Assets
	Locations
	Material Types
	Processes
	Products
	Suppliers
Hierarchy	Groups by Type

4.3.5 Customizing groups

Group types can be customized to suit your needs. This allows the groups to export properly to the Predictive Maintenance and Quality solution.

Two steps are required for the group customizations to take effect in the InfoSphere Master Data Management Collaborative Edition user interface:

1. Change the **Group Type** category in the **Groups by Type** hierarchy:
 - a. Select a **Group Type** in the **Groups by Type** hierarchy.
 - b. Provide a new **Code** and **Name** for the **Group Type**.
 - c. Save the changes.
2. Update the Group Hierarchy Lookup:
 - a. Go to the **Group Hierarchy Lookup table** (Figure 4-8).

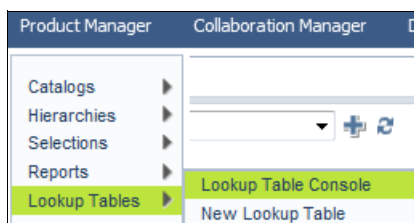


Figure 4-8 Group Hierarchy Lookup table

- b. Update the appropriate record to match the new **Group Type** code (Figure 4-9).

▼ Group Hierarchy Lookup (Entries 1 - 5 of 5) [Results 1 - 5 of 5]				
All <input type="checkbox"/>	Group Type Number	Hierarchy Name	Hierarchy Root Path	
<input type="checkbox"/>	1	Groups by Type	GROUPTYPE1	Edit
<input type="checkbox"/>	2	Groups by Type	GROUPTYPE2	Edit
<input type="checkbox"/>	3	Groups by Type	GROUPTYPE3	Edit
<input type="checkbox"/>	4	Groups by Type	GROUPTYPE4	Edit
<input type="checkbox"/>	5	Groups by Type	GROUPTYPE5	Edit

Figure 4-9 Group Hierarchy Lookup

4.3.6 Guidelines for managing master data

Consider the following guidelines when you use InfoSphere Master Data Management Collaborative Edition to import master data management:

- ▶ Assets:
 - Unassigned is OK.
 - Asset Categories/Hierarchy is OK, but not used by Predictive Maintenance and Quality.
 - Group assignments:
 - Can be assigned up to five groups from Groups by Type hierarchy.
 - Must be from a different Group Type.
 - Must be assigned to Group (Level 2), not Group Type (Level 1).
- ▶ Groups:
 - First level must be Group Type.
 - Second level should be Groups.
 - No catalog for Groups, and therefore only categories, no items.
- ▶ Locations:
 - First level must be Region (Location Type=Region).
 - Second level must be Country (Location Type=Country).
 - Third level must be State or Province (Location Type=State / Province).
 - Location items must only be defined under State / Province.
- ▶ Material Types
 - Unassigned is OK.
 - Categories/Hierarchy is OK, but not used by Predictive Maintenance and Quality.
- ▶ Processes
 - Unassigned is OK.
 - Categories/Hierarchy is OK, but not used by Predictive Maintenance and Quality.
- ▶ Products
 - Unassigned is OK.
 - Categories/Hierarchy is OK, but not used by Predictive Maintenance and Quality.
- ▶ Suppliers
 - Unassigned is OK.
 - Categories/Hierarchy is OK, but not used by Predictive Maintenance and Quality.

4.3.7 Configuring data export files

For the InfoSphere Master Data Management Collaborative Edition data to be processed into the Predictive Maintenance and Quality solution, the data export files must be sent to the Flat File API data input directory on the server that contains WebSphere Message Broker. The file transport mechanism can be any method that the user chooses. A sample implementation for Network File System (NFS) is provided here. The sample implementation allows InfoSphere Master Data Management Collaborative Edition exports to be immediately sent to the Message Broker server for processing.

To create the NFS share, complete the following steps:

1. On the WebSphere Message Broker server, perform these tasks:

- a. Ensure that NFS is configured to run:

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

- b. Share the data input directory for the Predictive Maintenance and Quality Flat File API by adding the following line to `/etc/exports` (create the file if it does not exist):

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

- c. Ensure that sufficient permissions are set on the Predictive Maintenance and Quality Flat File API input directory:

```
chmod 777 <wmb_fileapi_input_dir>
```

Permissions: The permissions shown in this step grant read and write privileges to all users and groups. A more secure configuration might be required. If a more secure approach is employed, ensure that users, groups, and permissions are in sync with those on the InfoSphere Master Data Management Collaborative Edition server so that NFS can operate properly.

- d. Restart the NFS service for the settings to take effect:

```
service nfs restart
```

2. On the InfoSphere Master Data Management Collaborative Edition server, complete these tasks:

- a. Ensure that the MDM data export directory exists. If it does not exist, create it.

```
mkdir <mdm_data_export_dir>
```

- b. Mount the remote Predictive Maintenance and Quality Flat File API input directory by using NFS:

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

3. Finally, test NFS sharing:

- a. Create a test file on the InfoSphere Master Data Management Collaborative Edition server:

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

- b. Check for the test file on the WebSphere Message Broker server:

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

If the file content is displayed, then NFS is working.

4.3.8 Running data exports

To run a data export, go to the MDM Reports Console as shown in Figure 4-10.

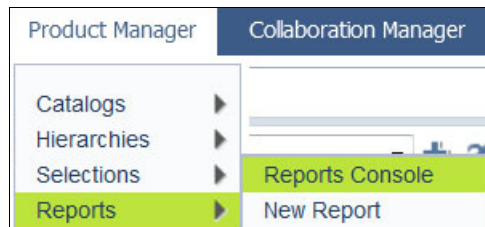


Figure 4-10 Reports Console menu selection

Then select the export and click the **Run** icon (green arrow) as shown in Figure 4-11.

Report Console					
	Name	Type	Schedule	Action	Delivery Location
Delete	IBM PMQ Asset Export	IBM PMQ Asset Export			IBM PMQ Export
Delete	IBM PMQ Group Export	IBM PMQ Group Export			IBM PMQ Export
Delete	IBM PMQ Location Export	IBM PMQ Location Export			IBM PMQ Export
Delete	IBM PMQ Material Type Export	IBM PMQ Material Type Export			IBM PMQ Export
Delete	IBM PMQ Process Export	IBM PMQ Process Export			IBM PMQ Export
Delete	IBM PMQ Product Export	IBM PMQ Product Export			IBM PMQ Export
Delete	IBM PMQ Supplier Export	IBM PMQ Supplier Export			IBM PMQ Export

Figure 4-11 Report Console

Data export files are written to `$PMQ_HOME/<mdm_data_export_dir>` (which is `$PMQ_HOME/data/export/mdm` by default).

4.4 Importing master data from IBM Maximo Asset Management

This section provides important details about importing master data into Predictive Maintenance and Quality from IBM Maximo Asset Management. The exact implementation steps vary depending on how Maximo Asset Management is installed and configured.

Maximo Asset Management is not bundled with the other components of the Predictive Maintenance and Quality solution. Therefore, this section will primarily benefit companies that are already using Maximo Asset Management in their enterprise.

4.4.1 Master data mapping

The information that is stored in Maximo Asset Management must be mapped into a different format before it can be read by Predictive Maintenance and Quality. The tables presented here give guidance for the mapping process. The assumption is that companies have extended this data model and that the default integration processes that are provided with

Predictive Maintenance and Quality must be extended to take advantage of this additional data.

Group_dim

The Group dimension (Table 4-26) is used for grouping the similar resources together. For example, all resources related to drilling assembly or painting assembly or organizations. These records provide classifications for resources. Up to five classifications are possible for each resource. The classifications can vary between uses of Predictive Maintenance and Quality.

Table 4-26 Mapping of Maximo attributes to attributes for group_dim table

Column name	Data type	Required or optional	Maximo object
group_type_cd	String	Required	"MXCLASSIFICATION"
group_type_name	String	Required	"Maximo Classification"
group_member_cd	String	Required	CLASSSTRUCTURE.CLASSSTRUCTUREID
group_member_name	String	Required	CLASSSTRUCTURE.DESCRPTION
language_cd	String	Optional	
tenant_cd	String	Optional	

Location

The location table (Table 4-27) contains the location of a resource or event, such as a room in a factory or a mining site. In Maximo Asset Management, this information is stored as a LOCATIONS object and in its associated SERVICEADDRESS object.

Table 4-27 Mapping of Maximo attributes to attributes for MASTER_LOCATION database table

Column name	Data type	Required or optional	Maximo object
location_cd	String	Required	SERVICEADDRESS.ADDRESSCODE
location_name	String	Required	SERVICEADDRESS.DESCRPTION
region_cd	String	Optional	SERVICEADDRESS.REGIONDISTRICT
region_name	String	Optional	SERVICEADDRESS.REGIONDISTRICT
country_cd	String	Optional	SERVICEADDRESS.COUNTRY
country_name	String	Optional	SERVICEADDRESS.COUNTRY
state_province_cd	String	Optional	SERVICEADDRESS.STATEPROVINCE
state_province_name	String	Optional	SERVICEADDRESS.STATEPROVINCE
city_name	String	Optional	SERVICEADDRESS.CITY
latitude	Decimal	Optional	SERVICEADDRESS.LATITUDE
longitude	Decimal	Optional	SERVICEADDRESS.LONGITUDE

Resource

Table 4-28 shows the mapping of asset resources from IBM Maximo Asset Management to the analytical data store in the Predictive Maintenance and Quality solution. Some asset resources exist in a hierarchy (for example, a truck is the parent resource of a tire). Asset information that is imported from Maximo Asset Management includes the asset type, classification, and location.

Table 4-28 Mapping of Maximo attributes to attributes for MASTER_RESOURCE database table

Column name	Data type	Required or optional	Maximo object
resource_cd1	String	Either of these is required: ▶ serial_no and model ▶ operator_cd	ASSET.ASSETNUM
resource_cd2	String	Required	ASSET.SITEID
resource_name	String	Required	ASSET.DESCRPTION
resource_type_cd	String	Required	
resource_sub_type	String	Optional	ASSET.ASSETTYPE
parent_resource_serial_no	String	Optional	ASSET.PARENT
parent_resource_model	String	Optional	ASSET.SITEID
parent_resource_operator_cd	String	Optional	
standard_production_rate	Float	Optional	
production_rate_uom	String	Optional	
preventative_maintenance_interval	Float	Optional	
group_dim_type_cd_1	String	Group codes are required, but a value of NA can be specified for a corresponding type and a member.	"MXCLASSIFICATION"
group_dim_member_cd_1	String		ASSET.CLASSSTRUCTUREID
group_dim_type_cd_2	String		
group_dim_member_cd_2	String		
group_dim_type_cd_3	String		
group_dim_member_cd_3	String		
group_dim_type_cd_4	String	Optional	
group_dim_member_cd_4	String	Optional	
group_dim_type_cd_5	String	Optional	
group_dim_member_cd_5	String	Optional	
location_cd	String	Required, but a code of NA can be specified	ASSET.SADDRESSCODE

4.4.2 Message flow for master data mapping

Integration Bus message flows are developed to manage master data loading from Maximo. Master data can be loaded from Maximo in two modes:

- ▶ **Batch Mode:** Exports existing master data as XML files and places them in the /maximointegrationin directory of the ESB node.
- ▶ **Real Time Mode;** When new master data is added in Maximo or existing master data is updated, it reaches the solution through web service outbound calls.

These message flows transform the Maximo data to fit the solution's master data structure, after which the transformed master data is processed through the flows explained earlier in this chapter.

A sample message flow is shown in Figure 4-12.

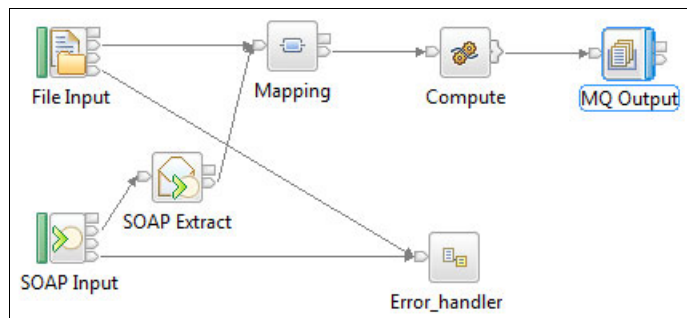


Figure 4-12 Message flow for loading master data exported from Maximo

The Integration Bus message flows have these components:

- ▶ **File node**
This node is used for reading and archiving the incoming files when operating in batch mode.
- ▶ **SOAP Input Node**
This node is used to retrieve the SOAP request for new or updated master data when operating in real-time mode.
- ▶ **SOAP Extract**
This node is used to extract the actual message from the SOAP envelope and pass it on for further processing.
- ▶ **Mapping node**
This node provides the mapping between Maximo objects to the format required by the solution.
- ▶ **Compute node**
This node creates an WebSphere MQ output message for the data that was transformed in the mapping node
- ▶ **MQ Output node**
This node puts the WebSphere MQ message created in the compute node onto the physical WebSphere MQ message queue.
- ▶ **Error handler subflow**
This subflow is started when an error is detected to write the error files.

4.4.3 Prerequisites

The following factors must be in place before the solution can be used for master data management:

- ▶ The required hardware and software are available.
- ▶ The server components of the Predictive Maintenance and Quality solution are installed by using the Predictive Maintenance and Quality Installer.
- ▶ The MQSI_FILENODES_ROOT_DIRECTORY environment variable is set to a proper existing location on the node where the Integration Bus is installed. The user placing the CSV files containing the master data files must be authorized to access this folder.
- ▶ The maximointegration, control, and error folders are present at the location defined by the MQSI_FILENODES_ROOT_DIRECTORY environment variable.

Guidance for completing these prerequisites is provided in the product documentation available here:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

4.4.4 Exporting data from Maximo Asset Management

Exporting master data from Maximo Asset Management involves tasks such as creating the external system and publish channel, and configuring the endpoint for the publish channel. These tasks are explained in the Predictive Maintenance and Quality Solution Guide available with other solution documentation at this location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>



Event mapping and processing

Event data comes from many sources and must be transformed into a usable format. The Predictive Maintenance and Quality (PMQ) solution uses IBM Integration Bus to perform these functions.

This chapter includes the following sections:

- ▶ Understanding events
- ▶ Event processing architecture
- ▶ Event processing in batch mode
- ▶ Event processing configuration parameters
- ▶ Parallel event processing
- ▶ Custom solution development

5.1 Understanding events

An event is defined as an occurrence of something, or an action that either has already occurred or is planned or projected to occur.

The IBM Predictive Maintenance and Quality solution is based on these events:

- ▶ **Actual**
Events that have occurred and serve as inputs for predictive analysis, such as temperature or pressure readings from sensors on drilling equipment
- ▶ **Planned**
Events that are expected to occur with values that fall within a specified range, such as a valve pressure reading with associated limits of tolerance
- ▶ **Forecast**
Events that are generated as a result of predictive analysis, such as estimates of a component's remaining life span, or warnings of imminent failure

5.1.1 Event processing

Figure 5-1 provides a high-level overview of event processing within the Predictive Maintenance and Quality solution.

The contents of each incoming external event are processed by the Integration Bus (shown here as Integration Bus) and stored in the IBM DB2 analytic data store database (shown here as PMQ Repository). The database tables consist of an event store to record the received events and tables containing the key performance indicators (KPIs) and profiles for the sources of events (the devices being monitored). The KPIs provide a history of device performance over time. The profiles show the current state of the device, but can also include information such as any recommended maintenance actions for the device that have been determined through predictive analysis.

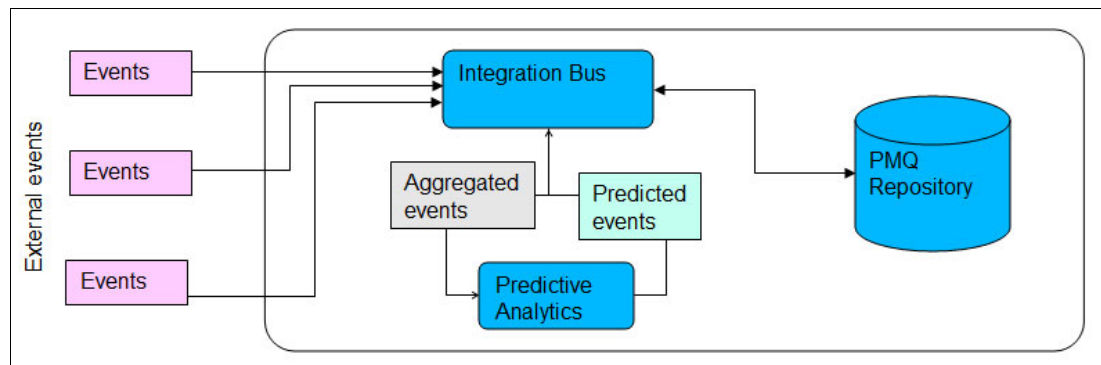


Figure 5-1 High-level overview of event processing

The event processing that occurs within Integration Bus can be divided into two stages. In the first stage, external events are received and mapped into (reformatted to fit) the input format that is required by the solution. Receiving and mapping external events requires a custom flow in WebSphere Message Broker. However, the solution includes a flow that accepts pre-mapped events from a CSV file. The mapped events are then placed on a different message queue for further processing. Events can also be placed on the queue directly as single events or as a group of multiple events that are processed together.

The processed events are inserted into the event store. Based on the information contained in the events, appropriate KPIs for the monitored device are updated, as is the historical record of all of the device's KPI values over time. In addition, the profiles that contain information about the current state of the monitored device are updated to reflect the most recent event readings.

Both the raw event data as well as the KPIs and profiles generated from that raw data can be used as inputs to analytical models that are used to generate predictive maintenance recommendations.

By processing events as they arrive and immediately updating the aggregated values in the KPI and profile tables in the analytic data store, the solution's dashboards and reports can immediately reflect the processed events.

ILS Technology's deviceWISE is an example of a product that can connect devices to the Predictive Maintenance and Quality solution. DeviceWISE reads sensor values from devices and has triggers to report when the values change. Each trigger causes a message to be sent to the solution, typically using a message queue. The message can be mapped to the solution's event format either in deviceWISE (before it is sent) or by implementing a custom message flow within the solution. For more information about integrating deviceWISE, see the PMQ Practitioners Restricted Community, which you can sign up for and access from the IBM developerWorks® Predictive Maintenance and Quality Community at this address:

<https://www.ibm.com/developerworks/connect/pmq>

5.1.2 Event data format

The Preventive Maintenance and Quality solution supports event data in two formats: Flat file input in CSV format and messages in tagged XML format. If the CSV format is used, the event is expected to be in a flat file that is placed in the `eventdata\in` folder. If the XML format is used, the message must be placed on the designated WebSphere MQ queue, `PMQ.EVENT.IN`.

Flat file event input

When events are supplied to the solution in CSV file format, each row in the file includes the measurement type, measurement value, resource type, source, and event type of the event. Each row also includes details about the device that is the source of the event (for example, name, serial number, location, and date time of occurrence). The source of every event is called a resource. The resource type can be either a machine (which the solution calls an asset) or a human operator (called an agent).

Each row of the input file can only define a single event, a single material, a single operator, and a single device. Therefore, any event that contains more than one of these elements requires more than one row. Resources that report multiple event types are captured by multiple rows.

Multi-row event

For resources that require more than one row to record multiple events, the optional `multi_row_no` column must be set so that the message flows can group all events that are related to the resource for the required processing. The information for the `multi_row_no` column is required and is expected to be provided in the input file. The `multi_row_no` column is optional and when absent indicates that a single event is being reported for the resource at the specified date and time.

The event information in every row of the CSV file has two parts:

- **Static:** Information that is common for all events coming from the resource. In Table 5-1, the first 10 rows (incoming_event_cd to tenant_cd) are common to all of the rows of a multi-row event.
- **Dynamic:** Time series data that contains the specific event information (in Table 5-1, the rows for measurement_type_cd to the end of the list contain dynamic information)

As an example, consider a drill bit resource that sends temperature, pressure, and RPM data at periodic intervals. The information about event type, source system process, production batch, location, serial number, and model number remains static, and is duplicated in three rows. The temperature, pressure, and RPM measurements are not duplicated.

Events must be premapped into the format that is shown in Table 5-1 to be loaded using the solution's message flows. Many of the fields are codes that reference values in master data tables.

Table 5-1 Event CSV file format

Field	Data type	Event or observation	Description
incoming_event_cd	String	Event	A unique code that identifies the event
event_type_cd	String	Event	The event type (for example Measurement, Alarm, Inspection)
source_system_cd	String	Event	The system generating the event
process_cd	String	Event	The production process that is related to the event
production_batch_cd	String	Event	The production batch that is related to the event
location_cd	String	Event	The location of the event
event_start_time	Datetime	Event	Time the event started
event_end_time	Datetime	Event	Time the event ended
event_planned_end_time	Datetime	Event	Time the event was planned to end
tenant_cd	String	Event	The organization associated with the event
operator_cd	String	Event	The operator associated with the event
model	String	Event	The model number of the device associated with the event
serial_no	String	Event	The serial number of the device associated with the event
measurement_type_cd	String	Observation	The measurement type determines how the event observation will be processed
observation_timestamp	Datetime	Observation	The time associated with the observation
value_type_cd	String	Observation	The type of observation (Actual, Planned, or Forecast)
observation_text	String	Observation	A description associated with the event

Field	Data type	Event or observation	Description
measurement	Float	Observation	A measurement associated with the event
material_cd	String	Observation	The material used for an event
multirow_no	Integer		For a multiple row event (more than one observation) use 1-to-n for each row of the event

Time stamps are represented in Coordinated Universal Time, referred to as UTC (for example, 2002-05-30T09:30:10Z). If events are sent to the Predictive Maintenance and Quality solution with time stamps that are not in UTC, they must be converted to UTC or event processing might fail.

The incoming events are mapped to the solution's standard event format. To better support batch processing, the format includes the ability to have groups of events.

Sample CSV file structure

Example 5-1 shows a sample for a sensor resource with Serial Number 13583 and Model Number M103. The example illustrates the order of the columns and the values that are associated with each column element of an event. All 14 rows belong to a single resource and are treated as a single event with 14 event observations or measurements.

Example 5-1 Sample event information in CSV format

```
incoming_event_cd,event_type_cd,source_system,process_cd,prod_batch_cd,location_cd,event_start_t
ime,event_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
1,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,TEMP,2013-01-01T00:00:00,ACTUA
L,,61.0562442,,1
2,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,INSP,2013-01-0100:00:00,ACTUAL
,,39.38833263,,2
3,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,LUBE,2013-01-0100:00:00,ACTUAL
,,24.23248517,,3
4,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,OPHR,2013-01-0100:00:00,ACTUAL
,,104.4750868,,4
5,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,PRS1,2013-01-0100:00:00,ACTUAL
,,43.03244035,,5
6,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,PRS2,2013-01-0100:00:00,ACTUAL
,,77.7851314,,6
7,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,PRS3,2013-01-0100:00:00,ACTUAL
,,105.1862696,,7
8,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,RELH,2013-01-0100:00:00,ACTUAL
,,77.73219341,,8
9,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,REPT,2013-01-0100:00:00,ACTUAL
,,61.95197408,,9
10,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,REPX,2013-01-0100:00:00,ACTUA
L,,31.02734505,,10
11,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,RPM,2013-01-0100:00:00,ACTUAL
,,11.13697433,,11
12,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,R_B1,2013-01-0100:00:00,ACTUA
L,,44.30237115,,12
13,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,M103,13583,R_F1,2013-01-0100:00:00,ACTUA
L,,43.16229167,,13
```

14,MEASUREMENT,SENSOR,,,Room1,2013-01-0100:00:00,,,PMQ,,,M103,13583,FAIL,2013-01-0100:00:00,ACTUAL,X101,1,,14

5.1.3 Schema definition of standard event format

Internally, the Predictive Maintenance and Quality solution processes events using the XML schema shown in Figure 5-2. Extensions of the solution that process external events from additional sources must map those events to this internal event format.

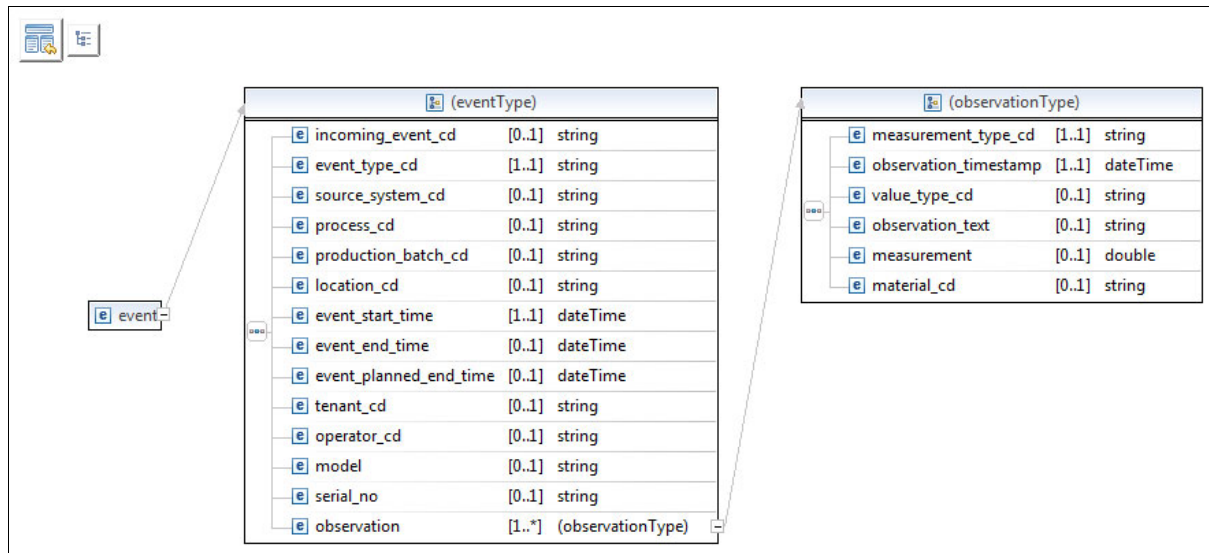


Figure 5-2 Schema that is used for validating the XML event messages

Sample XML message

Example 5-2 shows a portion of a sample XML message. The message belongs to a single resource and has 24 event observations.

Example 5-2 Sample XML message

```
<?xml version="1.0" encoding="UTF-8"?>
<event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../PMQEventDataLibrary/event.xsd">
  <incoming_event_cd>1</incoming_event_cd>
  <event_type_cd>MEASUREMENT</event_type_cd>
  <source_system_cd>SIGMAPLEX</source_system_cd>
  <process_cd>PA</process_cd>
  <production_batch_cd>WT2444_987</production_batch_cd>
  <location_cd>PA</location_cd>
  <event_start_time>2013-04-14T15:00:00</event_start_time>
  <event_end_time>2013-04-14T15:00:00</event_end_time>
  <event_planned_end_time>2013-04-14T15:00:00</event_planned_end_time>
  <tenant_cd>IT3</tenant_cd>
  <operator_cd>Operator 007</operator_cd>
  <model>BEDFORD</model>
  <serial_no>S42</serial_no>
  <observation>
    <measurement_type_cd>RPM</measurement_type_cd>
    <observation_timestamp>2013-04-14T15:00:00</observation_timestamp>
    <value_type_cd>ACTUAL</value_type_cd>
```



```

    <observation_text>WARNING</observation_text>
    <measurement>5000</measurement>
    <material_cd>FG</material_cd>
  </observation>
  <observation>
    <measurement_type_cd>FAIL</measurement_type_cd>
    <observation_timestamp>2013-04-14T15:00:00</observation_timestamp>
    <value_type_cd>ACTUAL</value_type_cd>
    <observation_text>X101</observation_text>
    <measurement>14.25</measurement>
    <material_cd>FG</material_cd>
  </observation>
  <observation>
    <measurement_type_cd>INSP</measurement_type_cd>
    <observation_timestamp>2013-04-14T15:00:00</observation_timestamp>
    <value_type_cd>ACTUAL</value_type_cd>
    <observation_text>Evt Typ 4</observation_text>
    <measurement>20</measurement>
    <material_cd>FG</material_cd>
  </observation>
  ...
  ...
</event>

```

Error reporting

Errors can occur while processing events. These errors can occur during the mapping of incoming data into the solution's required XML schema, or they can occur during the updating of the event, KPI, and profile tables in the analytic data store.

Any errors generated at different levels (for example, data validation, mapping, aggregation, or scoring) are captured and written into the `foundation.log` file.

5.2 Event processing architecture

The Analytics Solution Foundation (see 2.3, "Changes in Predictive Maintenance and Quality 2.0" on page 8) is a new component of the PMQ solution that reads events and produces recommendations. The goal of the Foundation architecture is to provide a reusable, configurable, and extensible component on which to base a number of analytic solutions. For more information about Foundation, see Appendix B of the *Predictive Maintenance and Quality Solution Guide*, which can be found with other solution documentation at this address:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

The Foundation provides the solution with an instance of an orchestration engine. The orchestration engine starts by associating each incoming event to an appropriate series of processing steps, collectively called an orchestration, that are defined in an XML file. Each step in the orchestration is performed by an adapter that works according to a predefined configuration that establishes how to process the event. Event processing steps vary depending on the type of event. An orchestration mapping is defined for each event type and these mappings determine which orchestration is performed for each incoming event.

Figure 5-3 provides a high-level overview of a possible event processing orchestration that shows the flow of an event into the orchestration engine. The contents of the event are processed and then stored in an event store. The contents of the event can be used to update KPIs and profiles. The KPIs provide a history of performance over time. The profiles typically contain only the current aggregated values. KPIs and profiles can also record information such as actions recommended by predictive models and the results of decision management calculations.

Foundation-related JAR files and configuration files are placed in the IBM Integration Bus (IIB) shared-classes directory (for example, /var/mqsi/shared-classes). If new orchestrations need to be defined or custom adapters need to be built, the orchestration XMLs and the JAR files that contain the custom adapters are likewise placed in this directory.

For more information about loading files from the IIB shared-classes directory, see the relevant section of the IIB Knowledge Center at this address:

http://www-01.ibm.com/support/knowledgecenter/SSMKHH_9.0.0/com.ibm.etools.mft.doc/bk58210_.htm%23bk58210_?lang=en

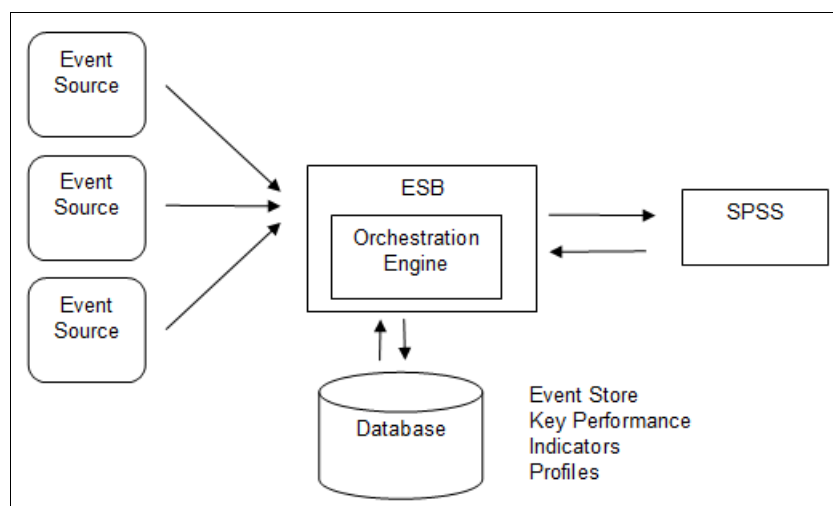


Figure 5-3 Event processing orchestration overview

As described above, the solution uses adapters that are provided with the Foundation to process events based on the steps outlined in an orchestration. Each step must specify one adapter_class to perform its associated task. An adapter might or might not require configurations. If an adapter requires configurations, the configurations must be provided under the step's adapter_configuration.

Several adapters are provided by the Foundation:

- EventStore adapter

The EventStore adapter is added to an orchestration to insert an event and any observations it contains into a database. If extra processing must be performed on raw events, a custom event adapter can be developed for that purpose.

Figure 5-4 shows a structured XML editor displaying an orchestration with a step that uses the EventStore adapter. The adapter_class value is the fully qualified name of the Java class that implements the adapter, in this case `com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter`.

Node	Content
?-? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	measurement
orchestration_cd	pmq.measurement
orchestration	
orchestration	
orchestration_cd	pmq.measurement
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
step	

Figure 5-4 Orchestration using the EventStore adapter

The EventStore adapter stores events and observations in the EVENT and EVENT_OBSERVATION tables, respectively.

Table 5-2 lists the elements of the EVENT table.

Table 5-2 EVENT table elements

Event element	Data type	Description
event_id	Integer	PMQ solution-generated event
event_start_time	Integer	Time when event occurred
event_end_time	Integer	Event end time
event_planned_end_time	Integer	Event planned end time
incoming_event_cd	String	Incoming event identifier
asset_id	Integer	Resource associated with the event
agent_id	Integer	Operator associated with the event
event_type_id	Integer	Event type
source_system_id	Integer	The source system generating the event
location_id	Integer	The location associated with the event.
process_id	Integer	The process associated with the event.
production_batch_id	Integer	The production batch associated with the event.
tenant_id	Integer	The tenant_id associated with the event

Table 5-3 lists the elements of the EVENT_OBSERVATION table.

Table 5-3 Table 5-3 EVENT_OBSERVATION table elements

Observation element	Data type	Description
event_id	Integer	PMQ solution-generated event identifier
observation_timestamp	Integer	Date time when observation was reported
observation_text	Integer	Measurement text
measurement	Integer	Measurement value
measurement_type_id	Integer	Measurement type associated with the event observation
value_type_id	Integer	Value type associated with the event observation
event_code_id	Integer	Event Code associated with the event observation
material_id	Integer	Material associated with the event observation
observation_date	Integer	Date when observation was reported
observation_time	Integer	Time when observation was reported
tenant_id	Integer	The tenant_id of the event observation

► Profile adapter

The Profile adapter is used to perform aggregation on raw events with calculations performed according to the adapter configuration that you define. Data is persisted into the KPI (RESOURCE_KPI, PROCESS_KPI) and profile (RESOURCE_PROFILE, PROCESS_PROFILE, MATERIAL_PROFILE) tables.

Figure 5-5 shows a structured XML editor displaying an orchestration that uses the Profile adapter. In this case, the fully qualified name of the Java class that implements the adapter is `com.ibm.analytics.foundation.adapter.profile.ProfileAdapter`.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
orchestration	
orchestration	
orchestration_cd	pmq.measurement
step	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xs:type	ns3:profile_adapter_configuration
observation_profile_update	
observation_selector	
profile_update_action	
profile_update_action	
profile_update_action	
profile_update_action	
observation_profile_update	
observation_profile_update	
observation_profile_update	

Figure 5-5 Orchestration using Profile adapter

A configuration for the Profile adapter consists of a series of profile updates. These profile updates define how the Profile adapter should calculate and update the values in profile tables (such as the `RESOURCE_KPI` and `RESOURCE_PROFILE` tables). The solution can be customized to calculate more profile values by defining new profile updates in a Profile adapter configuration.

Table 5-4 lists the elements of the `RESOURCE_KPI` table

Table 5-4 The elements of the `RESOURCE_KPI` table.

KPI element	Data type	Description
kpi_date	Date	The date for which the KPI is calculated. The time period 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 associated with the event.
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 that has an event code indicator of 1, the text from the event_observation_text is assumed to contain an event_code.
location_id	Integer	The location associated with the event.
process_id	Integer	The process associated with the event.
production_batch_id	Integer	The production batch associated with the event.

KPI element	Data type	Description
actual_value	Float	The actual value for this KPI table element. An important point to understand is that, for business intelligence reporting, this value is typically divided by the measure count. So even if the value is meant to be an average, this value should be a sum of the values from the event, and the measure count should be the number of events. This is done to support 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	This column provides the number of occurrences of the combination of resource, event code, location, process, and the profile variable.
current_indicator	Integer	An indication that this is the current row for a specific 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.

Table 5-5 lists the elements of the RESOURCE_PROFILE table.

Table 5-5 RESOURCE_KPI table elements

KPI element	Data 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, either 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 that has an event code indicator of 1, the text from the observation text is assumed to contain an event code.
location_id	Integer	The location associated with the event.
profile_date	Datetime	The recent date for which the profile is calculated.
prior_profile_date	Datetime	The date for which the profile was last calculated.
period_average	Float	The average value for the period.
period_min	Float	The minimum value for the period.
period_max	Float	The maximum value for the period.
period_total	Float	The total value for the period.
period_std_dev	Float	The standard deviation for the period.
period_msr_count	Integer	The number of events contributing to this profile for the current period.
prior_average	Float	The average value for the prior period.

KPI element	Data type	Description
prior_min	Float	The minimum value for the prior period.
prior_max	Float	The maximum value for the prior period.
prior_total	Float	The total value for the prior period.
period_std_dev	Float	The standard deviation for the period.
prior_msr_count	Integer	The number of events contributing to this profile for the prior period.
ltd_average	Float	The average value of the profile (obtained after computation based on metadata) for lifetime to date.
ltd_min	Float	The minimum value lifetime to date.
ltd_max	Float	The maximum value lifetime to date.
ltd_total	Float	The total value lifetime to date.
ltd_std_dev	Float	The standard deviation lifetime to date.
ltd_msr_count	Integer	The number of events contributing to this profile for the lifetime to date.
last_value	Float	The most recent value in the measurement column in the event_observation table for the resource.
tenant_id	Integer	The tenant_id of the profile_variable that is the source of this profile.

► Service adapter

The Service adapter is used to start external services. The Analytics Solution Foundation requires the definition of a Service adapter configuration to start specific services and prepare the necessary input and output for the service. The solution has custom service invocation handlers to define the actions to be performed when starting each service. If further action is required (for example, a service-generated event to record the result of starting the service), such an action can also be defined in the Service adapter configuration.

Figure 5-6 shows a structured XML editor displaying an orchestration with a step that uses the Service adapter. As shown in the figure, the fully qualified name of the Java class that implements the adapter is `com.ibm.analytics.foundation.adapter.b.ServiceAdapter`

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
orchestration	
orchestration	
orchestration_cd	pmq.measurement
step	
step	
step	
adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
adapter_configuration	
xsi:type	ns3:service_adapter_configuration
service_invocation_configuration	
service_invocation_configuration	
service_invocation_configuration	
event_selector	
service_invocation_action	
profile_row_selector	
profile_row_selector	
profile_row_selector	
profile_row_selector	
service_invocation	

Figure 5-6 Orchestration using Service adapter

5.2.1 Event loading and mapping

The Predictive Maintenance and Quality solution includes flows designed to support event processing.

Multi-row event loading from a CSV file

Event information is provided a CSV file that is placed in a designated folder. The MultiRowEventLoad message flow reads the message and generates an internal multiple-event XML message that is placed on the designated WebSphere MQ queue, PMQ.EVENT.IN, where the next flow, StdEventLoad, starts processing the event.

The MultiRowEventLoad message flow is shown in Figure 5-7.

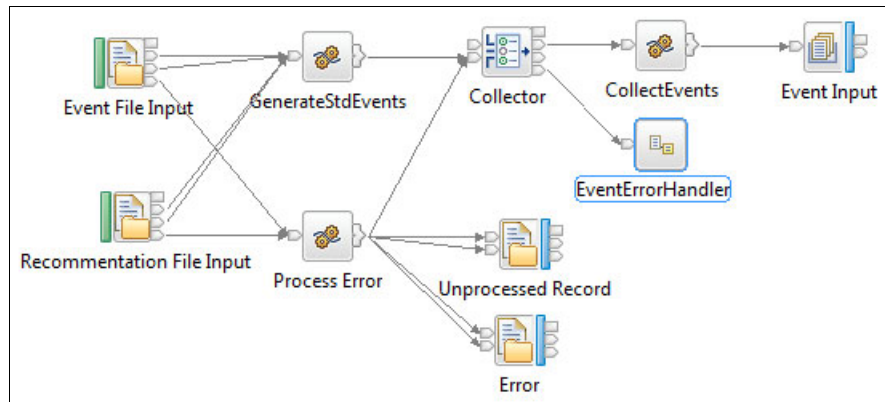


Figure 5-7 MultiRowEventLoad message flow

The MultiRowEventLoad message flow has the following nodes and subflows:

► **Event File Input node**

This node is used for reading the CSV-formatted events file at specified intervals from the eventdatain folder (in a typical PMQ installation, the /var/PMQ/MQSIFileInput/eventdatain folder).

► **Recommendation File Input node**

This node is used for reading the CSV-formatted events file at specified intervals from the integrationin folder (in a typical PMQ installation, /var/PMQ/MQSIFileInput/integrationin).

► **GenerateStdEvents node**

This compute node scans the rows in the CSV file, organizes the events by resource (device) based on tags included in the file, and then passes this grouped data to the collector node.

► **Collector node**

This node continues to collect row contents (events) until it completes the number of rows specified in the properties of the node and then provides that data in an array to the CollectEvents node, where the XML message is constructed.

► **CollectEvents node**

This compute node constructs the XML message from the multi-event data and then forwards the message to the EventInput node.

► **EventInput node**

This is the MQOutput node that is used to drop the messages on the input queue (PMQ.EVENT.IN).

► **EventErrorHandler subflow**

The EventErrorHandler subflow captures any errors that occur while generating the event XML message. These subflows manage errors. Error_handler captures any errors generated by the parsing of the incoming CSV file in terms of format.

► **Process Error node**

This compute node constructs the actual error message from the exception list.

- Error node

This node is used to write the error message in a file in the error folder (in a typical PMQ installation, /var/PMQ/MQSIFileInput/error).

- Unprocessed Record node

This node is used to write the unprocessed records in a file in the error folder (in a typical PMQ installation, /var/PMQ/MQSIFileInput/error).

Standard Event Handler

The next stage is to process the XML message that contains the event information. The XML message can be provided either by processing a CSV file containing multiple events or by allowing the sensors that monitor devices to directly push their events in XML format and then place them on the PMQ.EVENT.IN queue. The StdEventLoad message flow shown in Figure 5-8 is responsible for processing the XML messages. This message flow appears to be interrupted in the figure, but the flow continues from the Route To Label node to the appropriate label node (SetCredentials, SPSS, or Maximo).

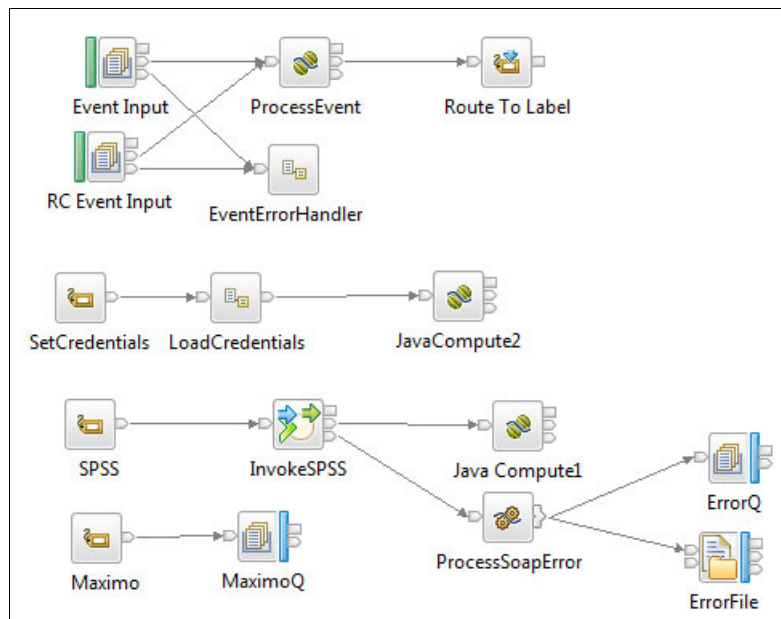


Figure 5-8 StdEventLoad message flow for processing events

The StdEventLoad flow has the following nodes and subflows:

- Event Input node

This node monitors the PMQ.EVENT.IN queue for incoming messages, which the node then forwards to the ProcessEvent Java compute node.

- RC Event Input node

This node monitors the PMQ.RC.EVENT.IN queue (typically used by integration flows to place messages containing analytical scores) for messages and forwards them to the ProcessEvent compute node.

- ProcessEvent node

This node is the heart of event management. It processes events based on the orchestration rules defined for the event type.

- ▶ **Route to Label node**
The flow continues from the Route To Label node to the appropriate label node (SetCredentials, SPSS, or Maximo) as determined by the message generated by the ProcessEvent node.
- ▶ **EventError Handler subflow**
The main purpose of this subflow is to capture errors and write the error information in the file in the error folder (in a typical PMQ installation, /var/PMQ/MQSIFileInput/error).
- ▶ **SetCredentials node**
This label node is triggered when the message emitted from the ProcessEvent node indicates to the Route To Label node that the flow should proceed to the SetCredentials node.
- ▶ **LoadCredentials subflow**
This subflow loads SPSS credentials from a properties file.
- ▶ **JavaCompute2 node**
This Java compute node saves SPSS credentials in shared memory for future reference.
- ▶ **SPSS node**
This label node is triggered when the message emitted from the ProcessEvent node indicates to the Route To Label node that the flow should proceed to the SPSS node.
- ▶ **InvokeSPSS node**
This SOAP request node starts the SPSS web service.
- ▶ **JavaCompute1 node**
This Java compute node processes the SOAP response that is returned by the SPSS web service.
- ▶ **ProcessSoapError node**
This compute node processes SOAP faults from the SPSS web service.
- ▶ **ErrorQ node**
This WebSphere MQ Output node writes the SOAP fault message to a designated WebSphere MQ queue.
- ▶ **ErrorFile node**
This file output node writes error messages to a file in the error folder (in a typical PMQ installation, /var/PMQ/MQSIFileInput/error).
- ▶ **Maximo node**
This label node is triggered when the message emitted from the ProcessEvent node indicates to the Route To Label node that the flow should proceed to the Maximo node.
- ▶ **Maximo Q node**
This WebSphere MQ output node writes an IBM Maximo work order creation message to a designated WebSphere MQ queue

Profile calculations

Profile calculations update both the KPI and profile tables for a resource, process, or material based on Profile adapter configuration.

The following list describes the behavior of the default profile calculations:

- ▶ **Measurement of Type**
This calculation determines the values of events with a particular measurement type (for example, the average, minimum, and maximum readings for a temperature event type).
- ▶ **Measurement of Type Count**
This is a count of the number of events that contain a particular measurement type.
- ▶ **Measurement in Range Count**
This is a count of the number of times the measurement value of an event falls within a specified high-low range
- ▶ **Measurement Text Contains Count**
This is a count of the number of times the text value of an event contains a specified string.
- ▶ **Last Date of Measurement in Range**
This is the most recent date on which an event occurred with a measurement value that fell within a specified high-low range.
- ▶ **Measurement Above Limit**
This is a count of the number of times the measurement value of an event falls above a specified limit.
- ▶ **Measurement Below Limit**
This is a count of the number of times the measurement value of an event falls below a specified limit.
- ▶ **Measurement Delta**
This is the change in a measurement value from one event to the next event
- ▶ **Custom Calculation**
The event processing flow can be modified to support more custom calculations.

For more information about these and other profile calculations, see the IBM Predictive Maintenance and Quality documentation available at this location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

5.3 Event processing in batch mode

The Predictive Maintenance and Quality solution supports event processing through .csv files. This approach can be used to process events in batch mode. Batch mode is primarily meant for loading historical events or bulk events. In batch mode, the solution uses these features:

- ▶ A Collector node to send events in groups of 100 to the Orchestration engine
- ▶ The Content Provider API provided by the Analytics Solution Foundation to support the processing of multiple events
- ▶ A pool of reusable EVENT and EVENT_OBSERVATION row objects provided by the Foundation, which avoids having to re-create these objects.

The Content Provider API provides good performance because it allows the orchestration engine to employ various optimizations when processing multiple events. This API enables setting the values in the event. This technique can be used to effectively process multiple events with a single call to the orchestration engine's processEvent method. Using a Content Provider API can allow a group of similar events to be processed more efficiently by allowing the orchestration engine to reduce network traffic to the database.

More details about processing events as a batch can be found in Appendix B of the *Predictive Maintenance and Quality Solution Guide*, which can be found with other solution documentation at this address:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

5.4 Event processing configuration parameters

The orchestration engine provided by the Analytics Solution Foundation supports various parameters for configuration and performance tuning. The parameters must be set before orchestration engine start.

5.4.1 Orchestration engine configuration parameters

Here is a list of orchestration engine configuration parameters:

- ▶ **ORCHESTRATION_DIR_KEY:** This parameter is used to define the directory that contains the orchestration XMLs. The path can be absolute or relative.
- ▶ **ORCHESTRATION_FILE_NAME_PREFIX_KEY:** This parameter is used to filter the orchestration XML files to load into the engine.
- ▶ **SOLUTION_DEFINITION_XML_KEY:** This parameter is used to define the directory that contains the orchestration XML files. The path can be absolute or relative.
- ▶ **ENABLE_XML_CONTENT_VALIDATION_KEY:** This parameter is used to enable and disable XML content validation of the solution definition and orchestration XML files. By default the value of this parameter is enabled.
- ▶ **ENABLE_PROFILE_ROW_CACHING_KEY:** This parameter is used to enable and disable the profile row cache. By default the value of this parameter is enabled.
- ▶ **ENABLE_PROFILE_UPSERT_BATCHING:** This parameter is used to enable and disable batch profile updates. When batch profile updates are enabled, all profile rows that are modified in the orchestration are persisted back to the database in a single batch update query. This happens at the end of the orchestration after all events are processed. When the profile row cache is disabled by the user, the value of this parameter is ignored and batch profile updates are disabled. By default the value of this parameter is enabled.
- ▶ **ENABLE_PROFILE_SELECT_BATCHING_KEY:** This parameter is used to enable and disable the batch selection of the profile rows that are required for a service invocation. Typically, a service invocation requires a set of profile rows as input. Some of these profile rows might not be present in the profile row cache. When batch selection is enabled, a single batch select query is used to retrieve all the additional profile rows that are required from the database. By default the value of this parameter is enabled.
- ▶ **ENABLE_PREPARED_STATEMENT_CACHING_KEY:** This parameter is used to enable and disable prepared statement caching. The caching of prepared statements avoids frequent query parsing and prepared statement creation. By default the value of this parameter is enabled.

5.4.2 Profile row pool configuration options

When profile row caching is enabled, the number of profile rows that are cached is determined by the profile row pool configuration. A pool of profile row objects is created and upon the request for a profile row during orchestration execution, an entry from the pool is selected and used. This configuration mechanism provides control over the number of active in-memory profile rows. If a profile row object isn't available in the pool and a new instance cannot be created due to the maximum active rows in memory cap being reached, the thread must wait until a profile row is returned back to the pool at the end of an orchestration.

The behavior of this pool can be fine-tuned by using these parameters:

- ▶ **PROFILE_ROW_POOL_MAX_ACTIVE:** This parameter is used to configure the maximum number of profile rows that can be active in the orchestration engine at any time. By default no limit is imposed. To enforce a limit, set this to a non-negative value.
- ▶ **PROFILE_ROW_POOL_MIN_IDLE:** This parameter is used to configure the target for the minimum number of idle profile rows to be maintained in the profile row pool. By default, the minimum number is 5.
- ▶ **PROFILE_ROW_POOL_MAX_IDLE:** This parameter is used to set a cap on the number of idle instances in the profile row pool. If the value of this parameter is set too low on heavily loaded systems, there might be profile row objects being destroyed and, almost immediately, new profile row objects being created. This is because the active threads momentarily return objects faster than they are being requested, causing the number of idle profile row objects to rise above the value of this parameter. The best value for this parameter for heavily loaded system varies, but the default (50) is a good starting point. Setting a negative value allows an unlimited number of idle instances (the number of profile rows in the pool never shrinks).

If any of these configuration parameters must be changed or more configuration parameters are needed, the contents of the ProcessEvent node of the Predictive Maintenance and Quality solution event flow StdEventLoad must be modified because this is where the values of these parameters are set.

5.5 Parallel event processing

Event processing can be made to 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. Because the rows in these tables are related to resources and measurement types, thread isolation must be achieved 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.

In the Predictive Maintenance and Quality solution, parallel event processing can be performed if the following conditions are met:

- ▶ Additional instances for the message flow are configured.
- ▶ Multiple copies of the message flow are created.
- ▶ Copies of the flow are made and no additional instances are enabled.
- ▶ Each copy is deployed on separate execution groups, rather than having all copies deployed in a single execution group, to help ensure that different processes run and performance is not impacted.
- ▶ The number of copies is limited by the system resources.

- ▶ Each copy of the flow has its own set of queues and folders to ensure that the process does not violate design principles.
- ▶ Each folder or queue contains unique resources, processes, and material, such that resources A, B, and C, if mapped to folder 1 and queue 1, are not also present in folder 2 or queue 2.

Figure 5-9 depicts these conditions. Parallel event processing is only supported for sensor events. It is not supported for inspection and warranty-related events.

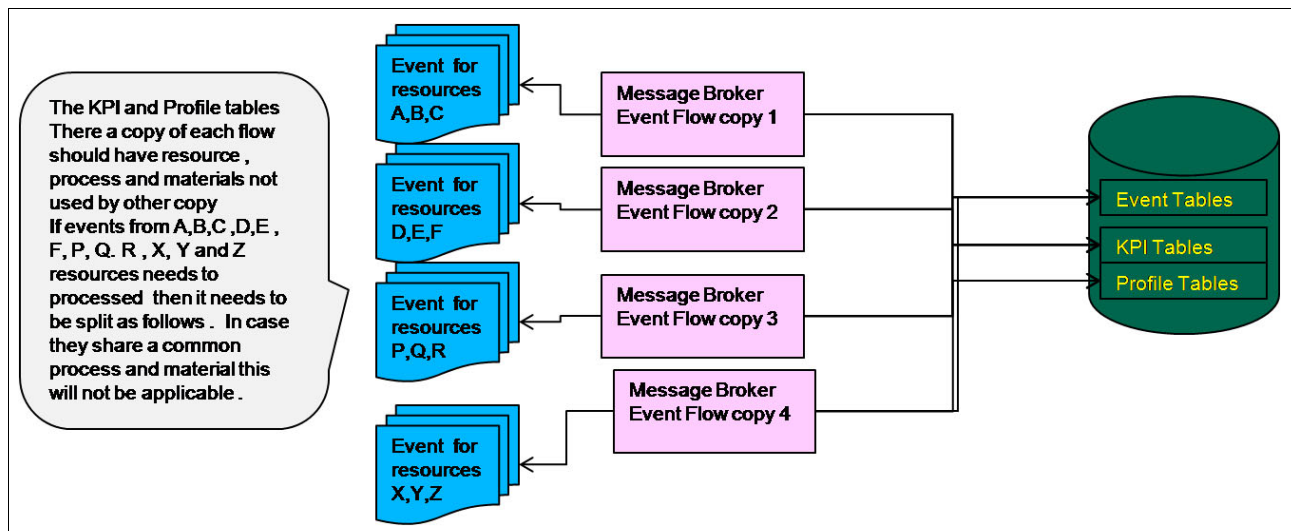


Figure 5-9 Parallel event processing showing required conditions

5.6 Custom solution development

The Predictive Maintenance and Quality solution can be extended to build custom solutions for specific customer needs. New measurement types and calculations can be configured in the solution using the configuration XML. Based on the analytical needs, custom models can be designed and added to the solution using the configuration XML. Customers can develop their own dashboards using the Cognos Business Intelligence Reporting capability of the solution. One such custom solution that was developed on top of existing Predictive Maintenance and Quality solution is the Predictive Safety Analytics (PSA) solution. For more information, see the PMQ Practitioners Restricted Community, which you can sign up for and access from the IBM developerWorks Predictive Maintenance and Quality Community at this address:

<https://www.ibm.com/developerworks/connect/pmq>



Sensor analytics

This chapter explains how the IBM Predictive Maintenance and Quality (PMQ) solution performs analytics on sensor-based data readings and includes an explanation of the various stages that are involved.

This chapter includes the following sections:

- ▶ Sensor readings and health score analytics
- ▶ Storing sensor data in PMQ
- ▶ Understanding sensor data
- ▶ Preparing sensor data for analysis
- ▶ Predictive modeling on sensor data
- ▶ Sensor model deployment and execution
- ▶ Sensor analytics reports

6.1 Sensor readings and health score analytics

A sensor is a device that observes changes in a device's state and provides a corresponding output value called a sensor reading. Typically, these readings involve a measurable parameter of an asset such as pressure or temperature. An asset in this context can be a small device, a large piece of manufacturing equipment, or even something bigger such as automobiles, aircraft, or entire buildings or office locations. Common examples include home-based devices such as smoke detectors, fire indicators, and carbon monoxide alarms, as well as speed sensors such as speedometers (used in automobiles), pitometer logs (ships), and pitot tubes (aircraft). For ease of reading, this chapter is presented using a simple example of a temperature sensor.

The PMQ solution can receive sensor readings, process them for interpretation, and perform various analytic techniques to produce maintenance recommendations for the monitored asset. Based on these readings, PMQ can calculate health scores that predict the performance of an asset or an asset-related process. The health score range is from 0 and 1, with larger values indicating better asset performance.

PMQ includes the Sensor Health Predictive Model. The Sensor Health Predictive Model is an analytical model that regularly monitors sensor readings to assess the health of the monitored asset based on both its current operating status and its historical sensor data. This data is stored in the solution's underlying database tables.

To use PMQ to calculate health scores, the sensor data must first be collected, processed, and prepared for storage and processing. At a high level, the sensor data is processed and loaded into the PMQ data store through the solution's Event Load flow (supported by IBM Integration Bus), a pre-built component that uses an orchestration definition file. See the previous chapters of this document to learn more about these PMQ components.

The next subsection shows how sensor readings are mapped to various database tables for further processing by the PMQ solution. After that, additional subsections explain how sensor readings are processed to produce health scores and predictions about asset maintenance.

6.2 Storing sensor data in PMQ

The solution processes sensor readings (also called observations) as events. An event is always associated with an observation and all observations are classified into specific measurement types. For example, a heat sensor observes changes in a device's temperature, so it sends events containing the measurement type Temperature.

The sensor data must be stored in the PMQ database tables, as expected by the Sensor Health Predictive Model. This model uses two tables: RESOURCE_KPI and MASTER_PROFILE_VARIABLE.

Each sensor reading is recorded as a specific measurement type and stored in the MASTER_PROFILE_VARIABLE table. In this particular example, the table holds the temperature value reported by the sensor.

Execution of the Sensor Health Predictive Model requires that all of the events within a single day (24-hour period) are aggregated and made available in a database. In PMQ, the RESOURCE_KPI table is used for this purpose. The same table is used to train the model (identify patterns in the data) and derive the health score. Any day with even a single failure event is considered a failure date in the context of the model.

6.3 Understanding sensor data

Figure 6-1 shows an IBM SPSS model that is used to understand the data before it is read for further execution by the predictive model that calculates the sensor health score. In Figure 6-1, the red boxes highlight the key nodes responsible for different aspects of the data.

- ▶ The output of the Data Audit node provides a comprehensive look at the data in an easy-to-read matrix that can be sorted and used to generate visual representations of the information.
- ▶ The output from the Table node helps in viewing the fields and values contained in the underlying database table, making this an easy way to inspect the data values.
- ▶ The output of the Statistics node gives summary statistics for individual fields and correlations between fields in the tables that are being used.

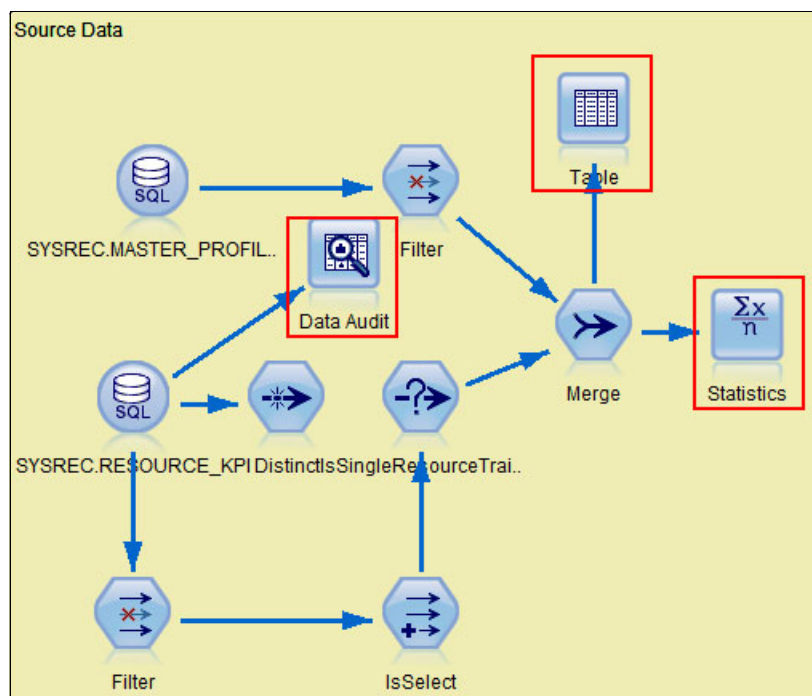


Figure 6-1 SPSS Modeler stream used for understanding the data

6.4 Preparing sensor data for analysis

The next step is preparing the sensor data for analysis. Each measurement type includes a base value and some additional values derived from additional calculations or inferences such as the number of times the temperature reading exceeded its upper and lower limits. Preparation of data in this context involves performing any required calculations on the raw data and then writing the output into a comma-separated value (CSV) file that serves as the input for the analytical model. Data preparation for batch mode operation is done using SPSS streams. For real-time events, preparation is done using the event processing flows in Integration Bus.

Table 6-1 describes SENSOR_HEALTH_DATA_PREP.str, the SPSS model involved in batch mode data preparation. The model reads the data from the underlying database, prepares them for computing the score and training the sensor health predictive model, and finally writes the output as a CSV file. This file is read by the Sensor Health Predictive Model that generates the health score.

Table 6-1 Sensor health data preparation stream

Stream name	SENSOR_HEALTH_DATA_PREP.str
Purpose	Extracts data from the solution's database tables and prepares the data to be used in the sensor health predictive model. All assets that have sufficient data in the database tables are selected for further processing. Aggregated sensor data for the selected assets (from solution's RESOURCE_KPI table) is exported to a CSV file.
Input	Source data contains values reported by sensors for each applicable measurement type
Output	Sensor_health_data.csv contains data for the eligible assets from the RESOURCE_KPI table to be used further for modeling and training of the health score model.

The model also must be trained to produce meaningful health scores, which requires that there be sufficient data from sensors sending FAIL measurement types (PMQ excludes assets that do not provide enough failure data as input sources for model training). This training data is logged in a separate file (Training_Eligibility_SensorAnalytics_Report.csv) in which assets are marked with either 1 (eligible) or 0 (not eligible). The assets that are excluded from training are marked with a health score of -1 and the Integration Bus message flows ignore them.

The SENSOR_HEALTH_DATA_PREP.str stream is shown in Figure 6-2.

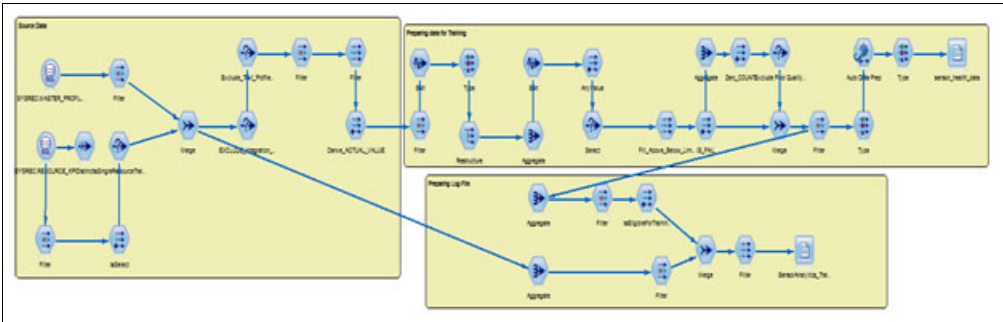


Figure 6-2 The SENSOR_HEALTH_DATA_PREP.str modeler stream

Training the models also involves Integration Bus message flows.

For real-time processing (as opposed to the batch mode operation just described), sensor events are processed through the regular Integration Bus event processing flows, as explained in Chapter 5, “Event mapping and processing” on page 61. More specifically, events are processed by an instance of an orchestration engine that runs steps defined for each event type. For sensor readings, the typical event type is MEASUREMENT. The orchestration rules governing this event type are defined in the PMQ_orchestration_definition_measurement.xml file that is included with the product.

Figure 6-3 shows a portion of the PMQ_orchestration_definition_measurement.xml file.

orchestration	(pool_configuration?, orchestration_cd, step?)
orchestration_cd	pmq.measurement
step	(adapter_class, adapter_configuration?)
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	(config_property?)*
step	(adapter_class, adapter_configuration?)
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	((config_property?)*, (event_profile_update*, observation_profile_update*))
step	(adapter_class, adapter_configuration?)
adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
adapter_configuration	((config_property?)*, (service_invocation_configuration?))

Figure 6-3 Orchestration definition XML for sensor events

The orchestration definition XML file for sensor events has sections dedicated for each step of the operation. The steps involve these adapters:

- ▶ Event Store Adapter: Raw events from sensors are placed in the EVENT and EVENT_OBSERVATION tables.
- ▶ Profile Adapter: Calculations are performed on the raw event data based on the Profile Adapter configuration that is defined in the orchestration XML file. For sensor analytics, the following calculations are performed and the results are written to the RESOURCE_KPI and RESOURCE_PROFILE tables:
 - Measurement of Type
 - Measurement Above Limit
 - Measurement Below Limit

The Profile Adapter configuration for Temperature (TEMP) is shown in Figure 6-4.

- ▶ Service Adapter: Aggregated data is sent to the solution's analytics engine to generate and retrieve the sensor health score and any related maintenance recommendations.

ns3:orchestration_definition	version="1.0" encoding="UTF-8"
xmlns:Q1	(event_orchestration_mapping*, orchestration*)
xmlns:ns3	http://www.ibm.com/asf
xmlns:xsi	http://www.ibm.com/asf/orchestration
xsischemaLocation	http://www.w3.org/2001/XMLSchema-instance
event_orchestration_mapping	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	(event_orchestration_key_cd, orchestration_cd+)
orchestration	(pool_configuration?, orchestration_cd, step?)
orchestration	(pool_configuration?, orchestration_cd, step?)
orchestration_cd	pmq.measurement
step	(adapter_class, adapter_configuration?)
step	(adapter_class, adapter_configuration?)
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	((config_property?)*, (event_profile_update*, observation_profile_update*))
xsitype	ns3:profile_adapter_configuration
observation_profile_update	(event_selector?, observation_selector, profile_update_action+)
observation_selector	(observation_field_value*)
table_cd	EVENT_OBSERVATION
observation_field_value	(field_name, value)
field_name	MEASUREMENT_TYPE_CD
value	TEMP
observation_field_value	(field_name, value)
profile_update_action	(profile_row_selector+, read_only_profile_row_selector*, shared_calculation_invocation_group_cd?, calculation_invocation*)
profile_row_selector	((table_cd shared_selector_cd), key_field_value*)
shared_selector_cd	RESOURCE_KPI
key_field_value	(profile_field_name, (value value_ref))
shared_calculation_invocation_group_cd	KPI_MEASUREMENT_OF_TYPE
profile_update_action	(profile_row_selector+, read_only_profile_row_selector*, shared_calculation_invocation_group_cd?, calculation_invocation*)
profile_row_selector	((table_cd shared_selector_cd), key_field_value*)
shared_calculation_invocation_group_cd	KPI_MEASUREMENT_ABOVE_LIMIT
calculation_invocation	(input_field_value*, update_field_value*)
profile_update_action	(profile_row_selector+, read_only_profile_row_selector*, shared_calculation_invocation_group_cd?, calculation_invocation*)
profile_row_selector	((table_cd shared_selector_cd), key_field_value*)
shared_calculation_invocation_group_cd	KPI_MEASUREMENT_BELOW_LIMIT
calculation_invocation	(input_field_value*, update_field_value*)

Figure 6-4 Profile Adapter configuration for the Temperature measurement type

6.5 Predictive modeling on sensor data

The next step is running a model to predict the health of the monitored asset. For temperature, this involves analyzing the transmitted values, their trends over time, and significant events such as when threshold limits are exceeded.

This section describes calculating key performance indicators (KPIs), computing the health score, generating maintenance recommendations, processing the generated health scores, and triggering work orders in IBM Maximo.

The health score is calculated using `SENSOR_HEALTH_COMBINED.str`, the stream deployed in the SPSS model server based on data from the `RESOURCE_KPI` table. This stream is shown in Figure 6-5. The Sensor Health Predictive Model performs three calculations: Measurement of Type, Measurement Above Limit, and Measurement Below Limit.

`SENSOR_HEALTH_COMBINED.str` helps to develop a model that can calculate the health score for an asset based on the measurement types that are received from the sensors. This is based on the SPSS Modeler auto classification model. The data that is prepared by `SENSOR_HEALTH_DATA_PREP.str` is used as the input and the health score for the asset is produced as the final output.

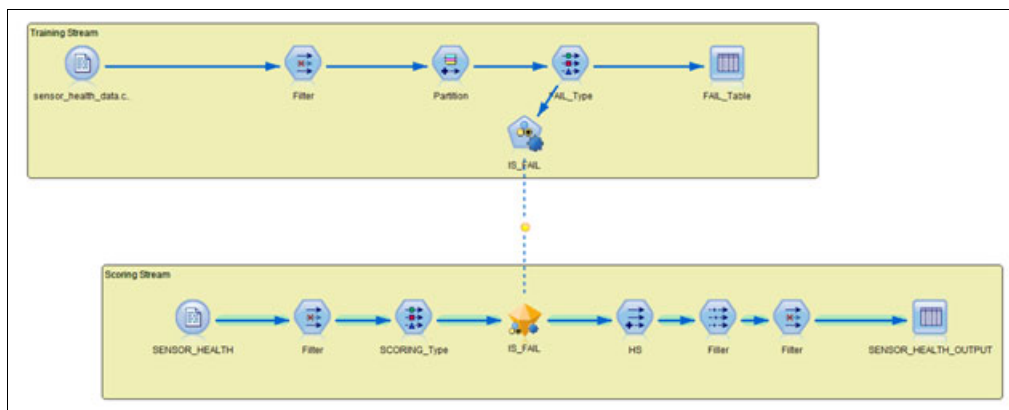


Figure 6-5 The `SENSOR_HEALTH_COMBINED` modeler stream

To obtain the health score, `SENSOR_HEALTH_COMBINED.str` is started by the Integration Bus message flows through the Service Adapter using a SOAP interface. The invocation parameters are defined in an XML file, as shown in Figure 6-6. The Service Adapter also stores the health score value in the database so that it is available for users to view on the dashboard as explained later in this chapter.

?? xml	version="1.0" encoding="UTF-8"
e ns3:orchestration_definition	
a xmlns:Q1	http://www.ibm.com/asf
a xmlns:ns3	http://www.ibm.com/asf/orchestration
a xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
a xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
e event_orchestration_mapping	
e event_orchestration_mapping	
e orchestration	
e orchestration_cd	pmq.measurement
e step	
e step	
e step	
e adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
e adapter_configuration	
a xsi:type	ns3:service_adapter_configuration
e service_invocation_configuration	
e event_selector	
e service_invocation_action	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	
e profile_row_selector	

Figure 6-6 Service configuration for sensor health score service

After obtaining the health score, the Integration Bus message flow starts the SPSS Analytical Decision Management service to obtain any asset maintenance recommendations that emerge from the analysis. The input for starting the `SENSOR_HEALTH_COMBINED` stream and the SPSS Analytical Decision Management service is prepared according to the configuration defined in the `profile_row_selector` section of the XML file.

An event is generated when the sensor health score service returns the value. The event, which is logged as the event type RECOMMENDED, carries the asset health score and any related recommendation. The event is processed through the standard event processing flow and the orchestration engine runs the steps that have been defined for the event type. The orchestration steps for this process are shown in Figure 6-7.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	recommended
orchestration_cd	pmq.recommended
event_orchestration_mapping	
orchestration	
orchestration_cd	pmq.recommended
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xsi:type	ns3:profile_adapter_configuration
observation_profile_update	
observation_profile_update	
observation_selector	
table_cd	EVENT_OBSERVATION
observation_field_value	
field_name	MEASUREMENT_TYPE_CD
value	HS
profile_update_action	
profile_update_action	
profile_update_action	
profile_update_action	
profile_update_action	
orchestration	

Figure 6-7 Orchestration definition XML for sensor health score events

The orchestration definition XML file for sensor health score events has sections dedicated to each step of the operation. The steps involve these adapters:

1. Event Store Adapter: Sensor health score events are written to the EVENT and EVENT_OBSERVATION tables.
2. Profile Adapter: Calculations are performed on the sensor health score data based on the Profile Adapter configuration. The results are written to the database tables. For sensor health score, three calculations are performed and the results are written to the RESOURCE_KPI and RESOURCE_PROFILE tables:
 - Measurement of Type
 - Measurement of Type Count
 - Measurement Text Contains Count

With the sensor health score and recommended maintenance actions received, the PMQ solution can also create work orders in the IBM Maximo enterprise asset management system. This feature is optional. If it is used, the work order creation service is started through the service invocation configuration defined in the orchestration XML, as shown in Figure 6-8.

?? xml	version="1.0" encoding="UTF-8"
e ns3:orchestration_definition	
a xmlns:Q1	http://www.ibm.com/asf
a xmlns:ns3	http://www.ibm.com/asf/orchestration
a xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
a xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
e event_orchestration_mapping	
e event_orchestration_mapping	
e orchestration	
e orchestration_cd	pmq.measurement
e step	
e step	
e step	
e adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
e adapter_configuration	
a xsi:type	ns3:service_adapter_configuration
e service_invocation_configuration	
e service_invocation_configuration	
e event_selector	
e service_invocation_action	
e profile_row_selector	
e service_invocation	
a service_cd	MAXIMO_WO_CREATE
a event_table_cd	EVENT
a event_context_variable_name	MAXIMO_WO_CREATE_EVENT
a handler_context_variable_name	MAXIMO_WO_CREATE_HANDLER
e input_field_value	
e input_field_value	
e input_field_value	
e input_field_value	
e input_field_value	
e input_field_value	
e input_field_value	
e service_invocation_configuration	

Figure 6-8 Profile Adapter configuration for sensor health score events

Sensor events are further processed by SPSS integration analytics models. For more information about these models, see Chapter 8, “Integration analytics” on page 113.

6.6 Sensor model deployment and execution

IBM SPSS Collaboration & Deployment Services (SPSS C&DS) Deployment Manager and IBM SPSS Analytical Decision Management are used to deploy the streams that are involved in sensor analytics.

6.6.1 Deployment

Sensor analytics models are deployed in the SPSS model server using IBM SPSS CaDS Deployment Manager. The deployable artifact, the `IBMPMQ_SENSOR_ANALYTICS.pes` archive, is provided with the PMQ solution and imported into SPSS C&DS Deployment Manager. Figure 6-9 shows this archive of the streams and jobs that are involved in sensor analytics, some of which have already been explained earlier in this chapter.

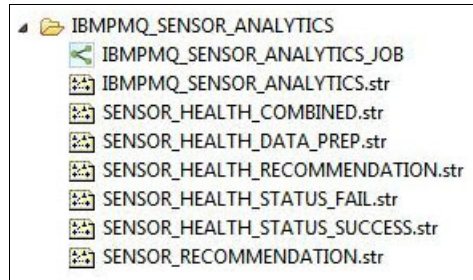


Figure 6-9 The `IBMPMQ_SENSOR_ANALYTICS.pes` archive

Table 6-2 lists the SPSS streams and jobs, and their purpose in the context of sensor analytics.

Table 6-2 SPSS streams and jobs involved in sensor analytics

Streams or jobs	Description
<code>SENSOR_HEALTH_DATA_PREP.str</code>	This stream retrieves and prepares sensor data for modeling and then exports the pertinent data in a CSV file that serves as the input for the downstream analysis.
<code>SENSOR_HEALTH_COMBINED.str</code>	This stream helps in training the models and also refreshes them for the scoring service.
<code>IBMPMQ_SENSOR_ANALYTICS</code>	This is an SPSS C&DS job that is used to start the <code>SENSOR_HEALTH_COMBINED</code> stream.
<code>IBMPMQ_SENSOR_ANALYTICS.str</code>	This stream is automatically generated for real-time scoring.
<code>SENSOR_RECOMMENDATION.str</code>	This stream generates recommendations based on the health score input.

In addition to the streams listed in Table 6-2, there are the `SENSOR_HEALTH_STATUS_FAIL` and `SENSOR_HEALTH_STATUS_SUCCESS` streams. These streams are used in the job to write the result status back into a file that helps Integration Bus to orchestrate and start the subsequent flows. The orchestration of starting the stream in the context of the `IBMPMQ_SENSOR_ANALYTICS` job is shown in the Figure 6-10.



Figure 6-10 `IBMPMQ_SENSOR_ANALYTICS` job flow

Deployment of the Sensor Health Predictive Model involves the parameters that are shown in Figure 6-11. When set to 1, the parameter IS_1_RES_TRAIN is used to train a single resource at a time, in which case the RESOURCE_ID needs to be passed along. When the value is 0, the RESOURCE_ID parameter is irrelevant because all of the resources are trained.

Parameters	Deployment	Execution	Globals	Search	Comments	Annotations
Name	Long name	Storage	Value			
IS_1_RES_TRAIN	Resource Training required	Integer	0			
RESOURCE_ID	Resource identifier	Integer	595			

Figure 6-11 Stream parameters

As stated earlier, the SENSOR_HEALTH_COMBINED stream helps train the models. It also refreshes the models for scoring. The stream has to refresh automatically each time, so the Refresh deployment option type is selected. This can be seen on the Deployment tab as shown in Figure 6-12.

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 6-12 Deployment tab with Model Refresh option selected

6.6.2 Recommendations

The sensor analytics application uses the optimization template provided with SPSS Analytical Decision Management. The SENSOR_RECOMMENDATION service is configured for scoring. This service is started to receive a recommendation for each resource, based on pre-configured business rules within SPSS Analytical Decision Management. This optimized recommendation is read in real time (as opposed to through batch processing) using a web services invocation from Integration Bus message flows.

The recommendations are communicated using these codes (also see Figure 6-13):

- ▶ HS101: Urgent inspection is needed. When SPSS Analytical Decision Management generates this recommendation code, the PMQ solution calls Maximo to create a work order.
- ▶ HS102: Non-urgent inspection is needed.
- ▶ HS103: No inspection is needed (asset performance is within acceptable limits).

Urgent Inspection	HS >= 0.7	HS101
Need Inspection	HS BETWEEN 0.4 and 0.7	HS102
Remainder		HS103

Figure 6-13 SPSS Decision Management Rules

The recommendations are prioritized based on a set of parameters as shown in Figure 6-14, which also shows the prioritization equation that is used. The prioritization depends on the probability of failure, maintenance cost of an asset, its expected lifespan, and the downtime involved in performing maintenance. In general, the more severe the risk, the higher the priority.

Campaign/Offer	Prob.to Fail	Maintenance Cost	Expected life time	Maintenance Downtime	Priority
Resource	HS		10		
HS101	HS	1000	10	20	High
HS102	HS	600	10	20	Normal
HS103	HS	200	10	20	Low

Prioritization Model (Value to be maximized)

Prioritization Equation

(Prob.to Fail * Maintenance Cost) - (Expected life time * Maintenance Downtime)

Figure 6-14 SPSS Decision Management Prioritization tab

6.7 Sensor analytics reports

There are two dashboards related to the Sensor Health Predictive Model. The details of the PMQ solution's reporting subsystem are provided in Chapter 13, "Additional reports" on page 163.

- ▶ Site Overview Dashboard
- ▶ Maintenance Overview Dashboard

6.7.1 Site Overview Dashboard

As the name implies, the Site Overview Dashboard provides a high-level summary of all monitored assets at all monitored sites. The dashboard shows the KPIs that have the greatest impact on asset maintenance. These include the KPIs for Health Score, Resource Count, Mean Time Between Failures (MTBF), and so on.

The Site Overview Dashboard (Figure 6-15) includes the following charts:

- Overview: Health Score Trend (bar chart)
- Overview: Health Score Contributor (pie chart)
- Overview: Incident and Recommendation Summary Report (bar chart)

This dashboard also provides a high-level summary of the health of all monitored assets.

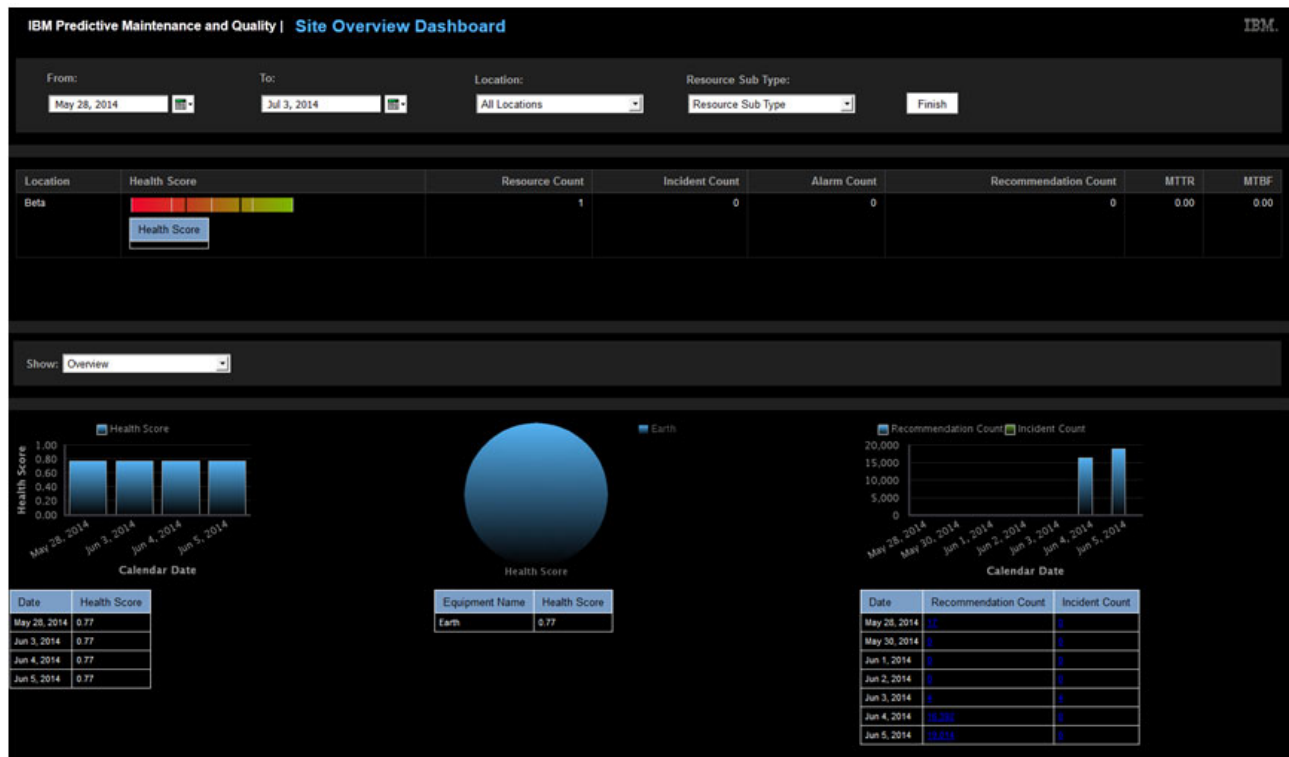


Figure 6-15 Site Overview Dashboard

6.7.2 Maintenance Overview Dashboard

Users can also view asset health score and asset recommendation details using the Maintenance Overview Dashboard (Figure 6-16). In the figure, the sensor health score is represented as HS.

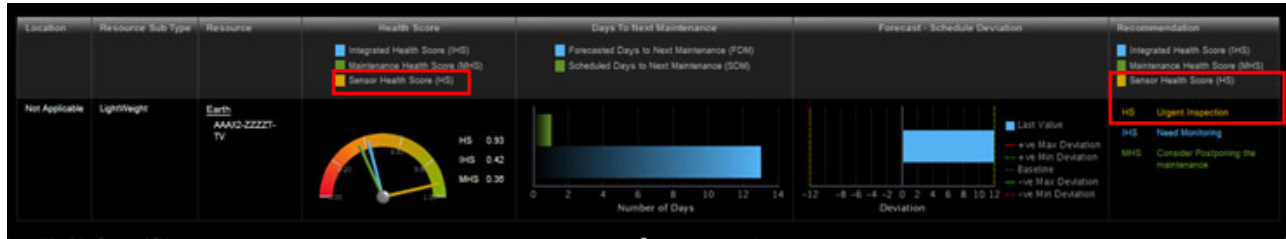


Figure 6-16 Maintenance Overview Dashboard



Maintenance analytics

This chapter explains how to use the IBM Predictive Maintenance and Quality solution to optimize maintenance and repair schedules at manufacturing plants and elsewhere. Traditional approaches to maintenance follow predetermined schedules, whether the asset truly needs maintenance then. By using the PMQ solution's analytical models, you can refine your schedules based on real-time data from the equipment being monitoring and historical data about previous repairs.

The solution's Maintenance Analytics module computes health scores for each asset based on their sensor readings and maintenance records in an enterprise asset management (EAM) system such as IBM Maximo. These scores help in forecasting upcoming maintenance work based on past work orders. The analytics process starts with a timer-controlled activity that periodically extracts maintenance-related data from the EAM system and loads it into the PMQ database.

This chapter includes the following sections:

- ▶ Data understanding of maintenance analytics
- ▶ Accessing and processing maintenance data
- ▶ Predictive modeling on maintenance data
- ▶ Manipulating maintenance analytics data
- ▶ Maintenance model deployment and consumption
- ▶ Maintenance analytics reports

7.1 Data understanding of maintenance analytics

Maintenance analytics analyzes historical data that contains both planned and breakdown maintenance work orders generated in an EAM system such as IBM Maximo. Two types of work orders are considered for maintenance:

- Planned maintenance (labeled MAINTENANCE in Maximo). This includes completed work orders and pending work orders expected to be completed in the future.
- Unplanned maintenance or breakdowns (labeled BREAKDOWN in Maximo).

To provide a business-level understanding of the maintenance data, the solution defines and computes key performance indicators (KPIs) and profile variables. Table 7-1 shows the profile variables that are most relevant to maintenance analytics:

Table 7-1 Profile variables related to maintenance analytic

Profile variable	Description
AMC	The number (count) of maintenance jobs already performed.
BC	The number (count) of breakdown episodes that resulted in maintenance activity.
SMC	The number (count) of maintenance jobs scheduled to be performed.

Maintenance analytics involves three primary PMQ database tables. The RESOURCE_KPI table holds aggregated KPI values for specific time periods such as one full day (24 hours). The MASTER_PROFILE_VARIABLE and MASTER_MEASUREMENT_TYPE tables are used to store and retrieve the measurement types and profile variable codes such as AMC, BC, and SMC.

Figure 7-1 shows a portion of the IBM SPSS user interface displaying how many of each profile are present in the data set for a resource. It also shows each profile count as a percentage of the whole and using color-coded proportion bars. In this example, 38.3% of all records in the underlying database relate to scheduled maintenance activities.

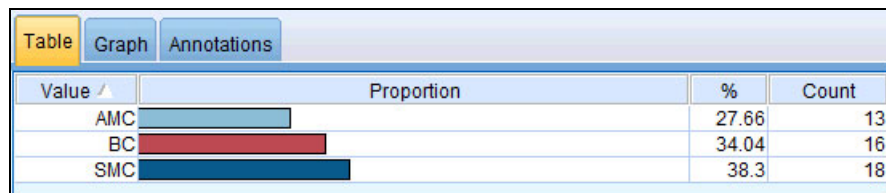


Figure 7-1 SPSS user interface showing distribution of profile variables in the data set

7.2 Accessing and processing maintenance data

Maintenance work orders (both planned and unplanned) must be accessible in the EAM system before analytics can be performed. There are two ways to access this data:

- In batch mode by exporting the data from the EAM system into a file system for access by IBM Integration Bus.
- In a real-time mode in which the EAM system sends the work orders to Integration Bus using a SOAP-based service call.

In either mode, Integration Bus work flows are used to map the work orders to PMQ events. After the work orders are mapped to PMQ events, they are processed by using the event processing flows.

The work orders are mapped to events by using the following transformation rules:

- ▶ **Planned maintenance (labeled MAINTENANCE in Maximo):**
Two events are generated for MAINTENANCE work orders. If the Scheduled Start field of the work order is populated, then an event with a measurement type set to SM (Scheduled Maintenance) is created. If the Actual Start field is populated (applies only to completed work orders), then an event with measurement type set to AM (Actual Maintenance) is created.
- ▶ **Unplanned maintenance or breakdowns (labeled BREAKDOWN in Maximo):**
One event is generated for each breakdown work order. The event has the measurement type BREAKDOWN.

The following subsections describe the workflow-level details of how these work orders are mapped to PMQ events, and how those events are processed by Integration Bus. For systems other than Maximo, the workflows described here still apply, but accessing the work orders can involve different mechanisms. For example, in EAM systems other than Maximo, the object names (such as MAINTENANCE or BREAKDOWN) will be different, and so the mechanisms for data export will differ as well.

7.2.1 Work order-to-event mapping (batch mode)

To convert historical MAINTENANCE and BREAKDOWN work orders into PMQ events in batch mode, the work orders are exported as XML files from Maximo to the /maximointegration directory of the enterprise service bus (ESB) node. The file naming pattern is SPEXTSYSTEM_SPWO_*.xml. More details can be found in the IBM Predictive Maintenance and Quality documentation at this location:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

The message flow that transforms these historical work orders into events is shown in Figure 7-2.

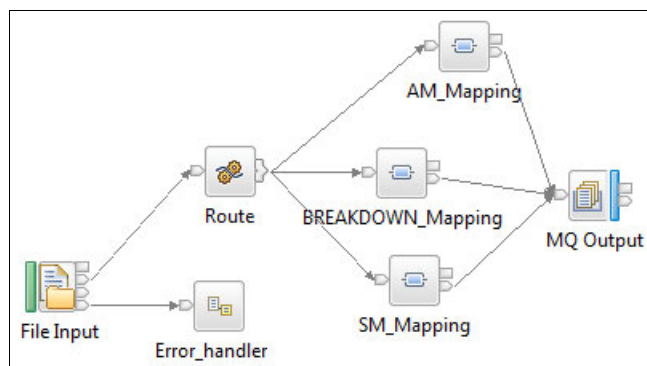


Figure 7-2 Work order mapping flow (batch mode)

The work order mapping flow (batch mode) depicted in Figure 7-2 involves these steps:

1. The File Input Node reads the XML files from the /maximointegration directory.
2. The File Input node reads the work order and routes it to the next node based on whether it is labeled BREAKDOWN (Breakdown Mapping node) or MAINTENANCE (SM Mapping

node if the SCHEDSTART field is not null or AM Mapping node if the ACTSTART field is not null). In each case, the mapping node creates an event message with an appropriate measurement type:

- Breakdown Mapping node: Measurement type Breakdown
 - SM Mapping node: Measurement type SM
 - AM Mapping node: Measurement type AM
3. The IBM WebSphere MQ Output node writes the event message to the PMQ.MAXIMO.BATCH.OUT message queue so it can be processed by the standard PMQ event processing flow.
 4. If errors are detected during this process, the Error Handler subflow is started to write the needed error files.

7.2.2 Work order-to-event mapping (real-time mode)

In real-time mode, new or updated MAINTENANCE and BREAKDOWN work orders are sent from the EAM system to the PMQ solution through Integration Bus in the form of a SOAP message. If Maximo is being used, a web service is started to send a SOAP message to Integration Bus (carrying the work order as the payload). If the EAM system is different, then this SOAP call must be custom-implemented using the capabilities of that system.

The message flow that transforms real-time work orders into events is shown in Figure 7-3

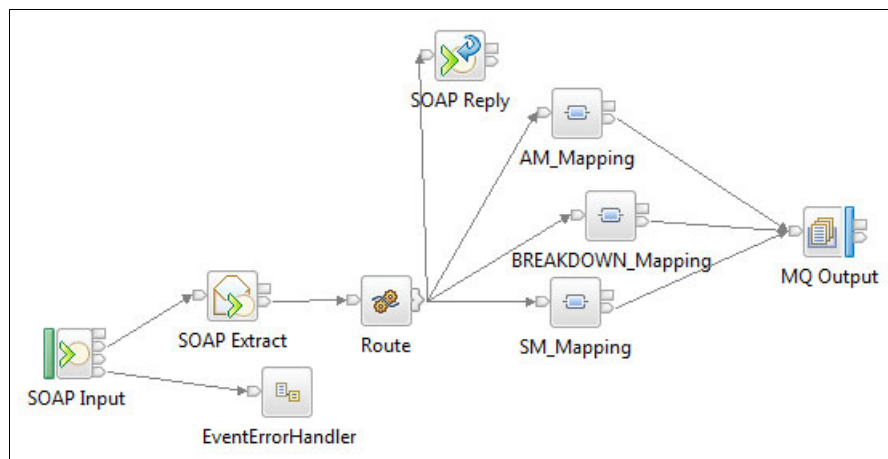


Figure 7-3 Work order mapping flow (real-time mode)

The work order mapping flow (real-time mode) depicted in Figure 7-3 includes these steps:

1. The SOAP Input node reads the SOAP message for the new or modified work order.
2. The SOAP Extract node extracts the work order message from the SOAP envelope.
3. The Route Compute node reads the work order data and sends it to the next node based on whether it is labeled BREAKDOWN (Breakdown Mapping node) or MAINTENANCE (SM Mapping node if the SCHEDSTART field is not null or its value has changed, or AM Mapping node if the ACTSTART field is not null or if its value has changed). In each case, the appropriate mapping node creates an event message with a measurement type of BREAKDOWN, SM, or AM.
4. The SOAP Reply node sends a SOAP response.

5. The MQ Output node writes the event message to the PMQ.MAXIMO.REALTIME.OUT message queue so the message can be processed by the standard PMQ event processing flow.
6. If errors are detected during this process, the Error Handler subflow is started to write the needed error files.

7.2.3 Maintenance event processing

When the historical maintenance data is available to the PMQ solution in the form of events, the events are processed according to the maintenance orchestration XML shown in Figure 7-4.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	mntrc
orchestration_cd	pmq.mntrc
orchestration	
orchestration	
orchestration_cd	pmq.mntrc
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
<!--	<step> <adapter_class> com.ibm.analytics.foundation.adapter.service.Servi

Figure 7-4 Orchestration definition XML for maintenance events

The definition XML for maintenance events has two orchestration steps:

1. The Event Store Adapter persists raw events into the EVENT and EVENT_OBSERVATION tables.
2. The Profile Adapter performs calculations on the raw event data based on the profile adapter configuration that is defined in maintenance orchestration XML. For maintenance events, the Measurement of Type Count calculation is performed and results are persisted to the RESOURCE_KPI and RESOURCE_PROFILE tables.

The Profile Adapter configuration for one measurement type, BREAKDOWN, is shown in Figure 7-5.

xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	maintenance
orchestration_cd	pmq.maintenance
event_orchestration_mapping	
orchestration	
orchestration_cd	pmq.maintenance
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xs:type	ns3:profile_adapter_configuration
observation_profile_update	
observation_selector	
table_cd	EVENT_OBSERVATION
observation_field_value	
field_name	MEASUREMENT_TYPE_CD
value	BREAKDOWN
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	KPI_COUNT
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	PROFILE_COUNT
observation_profile_update	
observation_profile_update	
orchestration	

Figure 7-5 Profile Adapter configuration for measurement type BREAKDOWN

7.3 Predictive modeling on maintenance data

After the maintenance work orders are retrieved and processed and the profile variables are computed, they are processed through IBM SPSS streams. This processing involves computing a Maintenance Health Score and predicting when future maintenance actions must be taken on the asset.

The appropriate SPSS stream is started by Integration Bus using a SOAP-based web service call.

7.3.1 Maintenance data processing

After maintenance events are processed, the associated data is available for SPSS to perform analytics. This involves the use of Integration Bus message flows to start SPSS jobs. The complete processing flow is shown in Figure 7-6.

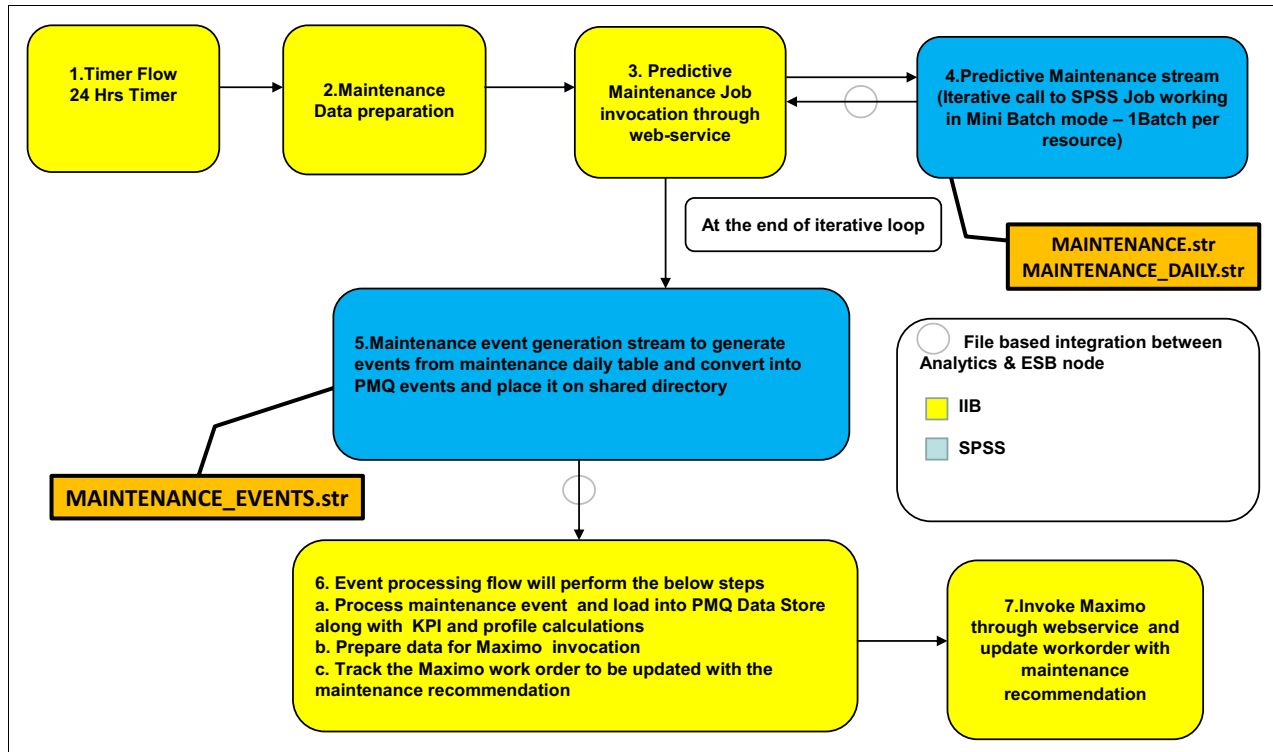


Figure 7-6 Maintenance data processing flow

The steps that are shown in yellow in Figure 7-6 are implemented using several Integration Bus message flows. The next subsections provide details about these flows.

Maintenance Timer message flow

The Maintenance Timer message flow (see Figure 7-7) runs at a scheduled time once each day to start maintenance analytics.

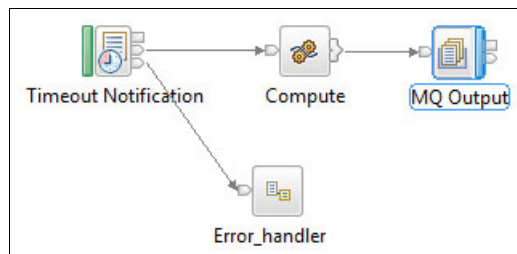


Figure 7-7 Maintenance Timer message flow

Maintenance Data Preparation message flow

The Maintenance Data Preparation message flow (see Figure 7-8) prepares the list of assets that are eligible for maintenance analytics and passes it to the Maintenance Job Integration flow. It also clears the tables where SPSS stores the analytical results (the MAINTENANCE_TRENDS, MAINTENANCE_DAILY, and MAINTENANCE_ADM_RECOMMENDATIONS tables).

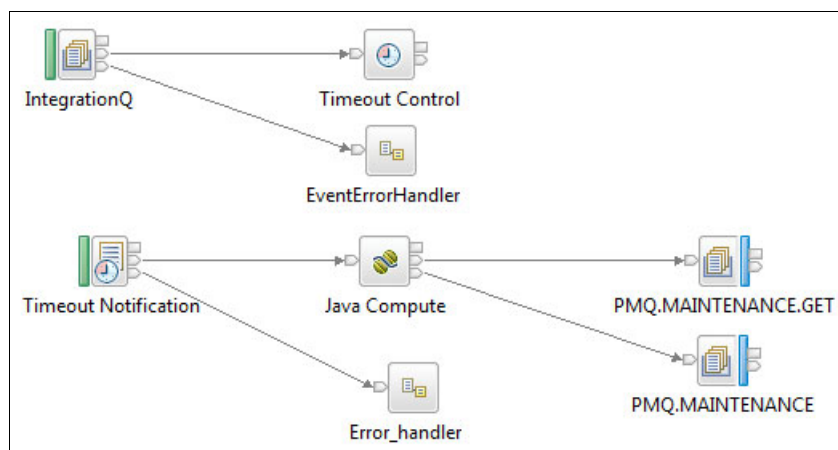


Figure 7-8 Maintenance Data Preparation message flow

Maintenance Job Integration message flow

Maintenance Job Integration message flow (see Figure 7-9) starts the SPSS Predictive Maintenance Job iteratively for each eligible asset. The SPSS stream runs in a batch mode for each asset and sends the status to the ESB node. After processing all assets, the flow starts the SPSS Maintenance Event Generation job.

The Maintenance Event Generation job generates events from the MAINTENANCE_DAILY table, converts them into the solution's required event structure, and places them in the shared directory of the Integration Bus node (/integration/in) for processing using the Event Processing flows explained in Chapter 5, “Event mapping and processing” on page 61.

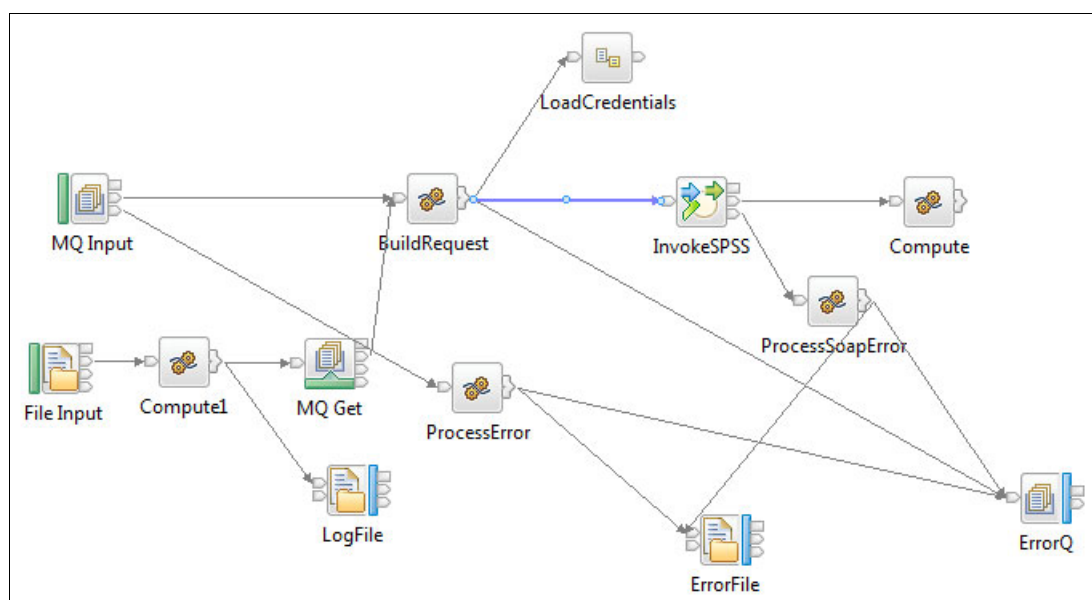


Figure 7-9 Maintenance Job Integration message flow

The results of maintenance analytics are called Maintenance Recommendation events. These events can contain several observations:

- ▶ Maintenance Health Score and recommendations such as to postpone or accelerate maintenance work
- ▶ Forecasted Days to Maintenance, along with the expected maintenance date
- ▶ Scheduled Days to Maintenance, along with the scheduled maintenance date
- ▶ Deviation between the forecasted and scheduled maintenance dates

The Event Processing flow processes Maintenance Recommendation events according to the maintenance orchestration XML shown in Figure 7-10.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	mntrc
orchestration_cd	pmq.mntrc
orchestration	
orchestration	
orchestration_cd	pmq.mntrc
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
--	<step> <adapter_class> com.ibm.analytics.foundation.adapter.service.Servi

Figure 7-10 Orchestration definition XML for Maintenance Recommendation events

The orchestration definition XML for Maintenance Recommendation events has three steps:

1. The Event Store Adapter persists Maintenance Recommendation events into the EVENT and EVENT_OBSERVATION tables.
2. The Profile Adapter performs calculations on the Maintenance Recommendation events based on the Profile Adapter configuration defined in the orchestration definition XML. For maintenance analytics, a Measurement of Type calculation is performed and the results are persisted to the RESOURCE_KPI and RESOURCE_PROFILE tables.

The Profile Adapter configuration for the measurement type MHS (Maintenance Health Score) is shown in Figure 7-11.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	mntrc
orchestration_cd	pmq.mntrc
orchestration	
orchestration	
orchestration_cd	pmq.mntrc
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xsitype	ns3:profile_adapter_configuration
observation_profile_update	
observation_selector	
table_cd	EVENT_OBSERVATION
observation_field_value	
field_name	MEASUREMENT_TYPE_CD
value	MHS
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	KPI_MEASUREMENT_OF_TYPE
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	PROFILE_MEASUREMENT_OF_TYPE
observation_profile_update	
observation_profile_update	
observation_profile_update	
!--	<step> <adapter_class> com.ibm.analytics.foundation.adapter.service.Ser

Figure 7-11 Profile Adapter configuration for the measurement type MHS

3. The Service Adapter is optional. You can enable this step (see Figure 7-12) if your assets are managed through IBM Maximo. After data aggregation is complete, Integration Bus starts the Maximo work order service to update the work order with the maintenance recommendation and forecasted maintenance date. The work order status is altered to CHANGE to indicate that there was an update.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
orchestration	
orchestration	
orchestration_cd	pmq.mntrc
step	
step	
step	
adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
adapter_configuration	
xs:type	ns3:service_adapter_configuration
service_invocation_configuration	
event_selector	
service_invocation_action	
service_invocation	
service_cd	MAXIMO_WO_UPDATE
event_table_cd	EVENT
handler_context_variable_name	MAXIMO_RECOMMENDATION
input_field_value	
input_field_value	
input_field_value	
input_field_value	
input_field_value	
input_field_value	
input_field_value	

Figure 7-12 Service Adapter configuration for assets managed in Maximo

7.3.2 SPSS stream for predicting maintenance schedules

MAINTENANCE.str is the predictive model that computes the Maintenance Health Score and the number of days until maintenance activity is needed (achieved using time series modeling). Table 7-2 provides details about the MAINTENANCE stream.

Table 7-2 MAINTENANCE stream

Stream name	MAINTENANCE.str
Purpose	This stream predicts the number of days before maintenance will be needed on an asset. These forecasts are then used to compute continuous health scores about asset performance as a function of the maintenance-related forecasts.
Input	The profiles for the actual, planned, and scheduled maintenance dates extracted from the work orders in the EAM system.

Stream name	MAINTENANCE.str
Output	<p>The output of this stream takes two forms:</p> <ul style="list-style-type: none"> ▶ Number of days until the maintenance is next needed on the asset ▶ Current health score of the asset (one score is computed each day) <p>The output is written to the MAINTENANCE_TRENDS table.</p>

7.4 Manipulating maintenance analytics data

To improve reporting performance, and ensure fast data refreshes and an optimal user experience, certain data manipulations are performed by SPSS batch jobs, which can be run during off-peak hours. This is done by using the MAINTENANCE_DAILY stream, which writes the new data in the database for easy reporting.

Table 7-3 describes the MAINTENANCE_DAILY STREAM and its outputs.

Table 7-3 MAINTENANCE_DAILY stream

Stream	MAINTENANCE_DAILY.str
Purpose	This is a data preparation stream. It runs after MAINTENANCE.str calculates the health score and next required maintenance dates. The stream converts predictions from MAINTENANCE_TRENDS table into the format required by the Maintenance Overview Dashboard and stores them in the MAINTENANCE_DAILY table.
Input	The input data consists of all records present in the MAINTENANCE_TRENDS table for all measured days.
Output	The current day's data calculations are sent to the MAINTENANCE_DAILY table.

MAINTENANCE_EVENTS.str reads data from MAINTENANCE_DAILY table and converts them into the solution's required event format. Table 7-4 describes of the purpose and outputs of the MAINTENANCE_EVENTS stream.

Table 7-4 MAINTENANCE_EVENTS stream

Stream name	MAINTENANCE_EVENTS.str
Purpose	This is a post-modeling data preparation stream that converts the data in the MAINTENANCE_DAILY table into the solution's required event format.
Input	The input data source consists of all records present in the MAINTENANCE_DAILY table.
Output	Output consists of a CSV file with the MAINTENANCE_DAILY data in a format that can be used by Integration Bus message flows.

The SPSS streams described above are started by the Integration Bus message flows explained earlier.

7.5 Maintenance model deployment and consumption

The developed models are deployed and read by using SPSS Collaboration and Deployment Services (SPSS C&DS) Deployment Manager and SPSS Analytical Decision Management.

7.5.1 Deployment by using SPSS C&DS Deployment Manager

At run time, the Maintenance model uses parameters that are configured at the time of deployment. If these parameters are not assigned values at the time of deployment, a default value will be used.

The deployment-level model parameters are shown in the SPSS interface shown in Figure 7-13. Some of these parameters are directly involved in model execution whereas others are required by downstream applications and so are carried forward by the model. Each parameter has a default value that will be used in the absence of a user-provided one.

Parameters	Deployment	Execution	Globals	Search	Comments	Annotations
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 7-13 SPSS model parameters

Table 7-5 provides details of the streams and jobs that are involved in the process.

Table 7-5 SPSS streams and jobs used in maintenance analytics

Streams and jobs	Description
MAINTENANCE.str	This is the main stream that calculates the Maintenance Health Score and forecasts the next anticipated maintenance date.
MAINTENANCE_DAILY.str	This stream generates the maintenance calculations for each day.
MAINTENANCE_RECOMMENDATIONS.str	This stream helps to generate the maintenance recommendations.
IBMPMQ_MAINTENANCE_ANALYTICS_JOB	This is an SPSS C&DS job used to start the maintenance streams described above.
MAINTENANCE_EVENTS.str	This stream creates a CSV file that helps in plotting the data on a dashboard.

In addition to these streams and jobs, also present are the MAINTENANCE_STATUS_FAIL.str and MAINTENANCE_STATUS_SUCCESS.str streams, which are used to write the results to a file that helps Integration Bus orchestrate the subsequent flows.

7.5.2 Recommendations from SPSS Analytical Decision Management

Based on the scores and data received from MAINTENANCE.str, you can determine whether any particular maintenance must be postponed or accelerated.

As shown in the Figure 7-14, the output of MAINTENANCE.str is read by SPSS Analytical Decision Management to produce a text-based result based on business rules. These recommendations indicate whether the scheduled maintenance activity needs to be advanced, postponed, or something else. The interface shows the actions (such as prepone or keep as planned) along with the rules based on deviations.





<input type="checkbox"/> <u>Prepone_Maintenance_Dev_LT_-100</u>	2005
 DEVIATION_PERCENT < -100	
<input type="checkbox"/> <u>Maintenance_as_planned_bet_0_10</u>	3001
 DEVIATION_PERCENT BETWEEN 0.0 and 10.0	
<input type="checkbox"/> <u>Maintenance_as_planned_bet_-10_0</u>	3002
 DEVIATION_PERCENT BETWEEN -10.0 and 0.0	
<input type="checkbox"/> <u>No Forecast Available</u>	4001
 FORECASTED_DAYS IS NULL	

Figure 7-14 SPSS Analytical Decision Management rules

7.6 Maintenance analytics reports

The PMQ solution includes numerous reports that can give insight into the analyzed maintenance data and the corresponding recommendations.

7.6.1 Maintenance Overview Dashboard

The Maintenance Overview Dashboard shows reports and charts involving the Sensor Health Score, Maintenance Health Score and Integrated Health Score for each asset and location.

The Sensor Health Score is a near real-time value that is calculated from sensor readings. The Maintenance Health Score is calculated from maintenance logs. The Sensor Health Score and the Maintenance Health Score are combined to give the Integrated Health Score.

In the charts displayed in the maintenance dashboard (see to Figure 7-15 on page 111), you can set the following filters:

- ▶ Location
- ▶ Health Score
- ▶ Recommendation
- ▶ Absolute Deviation per cent
- ▶ Forecasted Days to Next Maintenance
- ▶ Scheduled Days to Next Maintenance
- ▶ Event Code



Figure 7-15 Maintenance Overview Dashboard

You can also click Summary to see things such as the total number of resources and total counts of recommendations for each of the resources (Figure 7-16).



Figure 7-16 Maintenance Overview Dashboard, Summary view

7.6.2 Maintenance Advance Sorting report

You can click Advance Sorting to see an Advance Sorting report (Figure 7-17). The report displays the same data that is shown in the main report, but presents it in a tabular format that is enabled for sorting.

IBM Predictive Maintenance and Quality Maintenance Advance Sorting										
Prompt Selection										
Location	Resource Sub Type	Resource	Resource Code	Maintenance Health Score	Sensor Health Score	Integrated Health Score	Forecasted Days to Next Maintenance	Scheduled Days to Next Maintenance	Deviation	Resource Deviation
Not Applicable	LightWeight	Earth	AAAX2-ZZZZT-TV	0.73			5	8	-3	3

Figure 7-17 Maintenance Advance Sorting report

Click a particular resource in the Resource column helps the user drill through to the Maintenance Health and Failure Detail Report for the resource (Figure 7-18).

The user can include or exclude events from the report:

- ▶ Breakdown maintenance
- ▶ Planned maintenance
- ▶ Forecasted maintenance
- ▶ Scheduled maintenance

Each included event appears as a bar on the chart. The bar indicates the date on which the event occurred. The y axis shows the health score. The x axis shows the date on which the health score was computed. 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 score shown for the current date is the current health score.

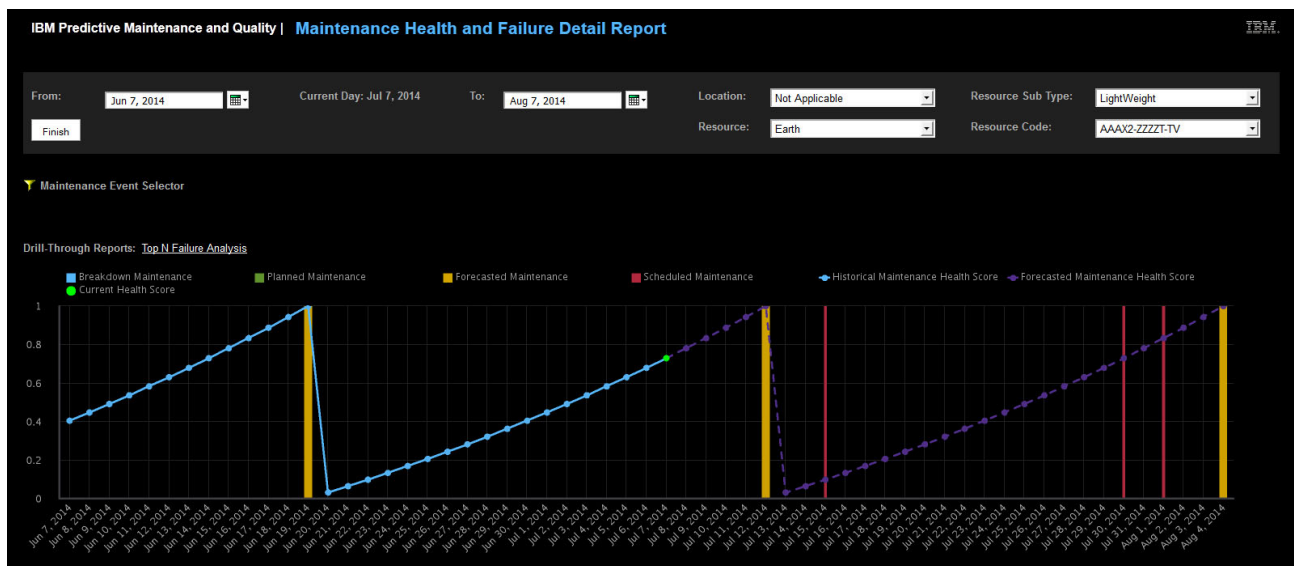


Figure 7-18 Maintenance Health and Failure Report



Integration analytics

A company can have multiple predictive models for a single resource, each producing different recommendations. This flood of recommendations can lead to suboptimal decisions or even decision paralysis. Ultimately, one recommendation must be chosen for action.

This chapter explains how to perform integration analytics with the IBM Predictive Maintenance and Quality (PMQ) solution. The solution's Integration Analytics module can read the output from multiple analytical models to produce a unified maintenance recommendation for the resource. This unified recommendation takes the form of an integrated health score.

The chapter includes the following sections:

- ▶ 8.1, “Business understanding of integration analytics” on page 114
- ▶ 8.2, “Data understanding of integration analytics” on page 114
- ▶ 8.3, “Data preparation for integrated analytics” on page 115
- ▶ 8.4, “Predictive modeling for integrated analytics” on page 116
- ▶ 8.5, “Deploying integrated health score models” on page 119
- ▶ 8.6, “Integrated Analytics Report” on page 123

8.1 Business understanding of integration analytics

To grasp the importance of integration analytics, consider fictitious Company D, a modern manufacturing company that uses advanced analytical systems to monitor the health of its machinery and predict appropriate maintenance schedules and repairs. Yet even with advanced predictive solutions at its disposal, the company still struggles to make effective maintenance decisions.

Company D has procured a text analytics system from Vendor A, a sensor-based predictive system from Vendor B, and a separate sensor-based content management system (also from Vendor B). In addition, the company has more limited predictive systems provided by the makers of its industrial machinery.

Each system generates different analytical results and sometimes conflicting recommendations. Company D's maintenance managers are often unable to choose which recommendations to follow and resort to performing extra manual inspections of problem machinery. This defeats the purpose of using sophisticated predictive analytics.

The problem is that in many cases, the most accurate result does not emerge from a single analytic source and instead requires merging results from multiple sources, sometimes using complex, non-linear mathematical computations involving different weighting criteria. This is what integration analytics is all about. The result is a single recommendation that adds clarity and enhances the speed and accuracy of decision making.

The Integration Analytics module in PMQ helps reconcile conflicting recommendations by using complex machine learning and artificial intelligence algorithms.

Similar to all analytical models, the one for integrated analytics involves the use of IBM SPSS streams to train the model, calculate the needed score, and evaluate the results. Historical data must be prepared before the scoring-related stream can be trained. Details about these steps are provided in this chapter.

8.2 Data understanding of integration analytics

To fully understand integration analytics, one must understand the underlying data and output from sensor analytics (see Chapter 6, “Sensor analytics” on page 83) and maintenance analytics (see Chapter 7, “Maintenance analytics” on page 97). Values from these related analytical processes, such as Maintenance Score, Sensor-Based Health Score, and Forecasted Maintenance Days, are populated into the RESOURCE_KPI and MASTER_PROFILE_VARIABLE tables (see Chapter 4, “Master data loading” on page 25).

There are many ways to view and gain a deeper understanding of the data in these tables. Figure 8-1 shows one of these methods, a data audit window in IBM SPSS Modeler. Here you can view summary statistics, histograms, and distribution graphs to learn more about the data.

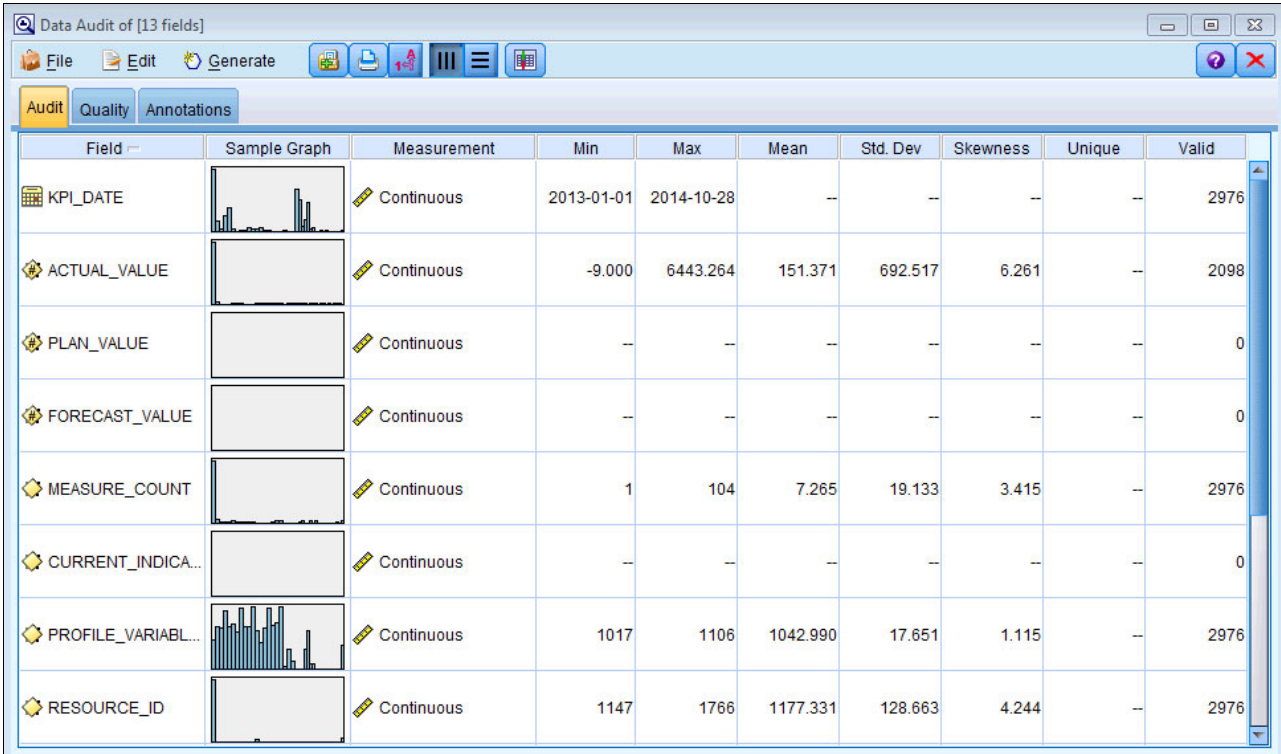


Figure 8-1 Using SPSS Modeler to view data in the RESOURCE_KPI and MASTER_PROFILE_VARIABLE tables

8.3 Data preparation for integrated analytics

The ready-to-use integrated analytics module in PMQ calculates a combined health score for an asset based on the separate scores that emerge from the solution's maintenance and sensor analytics modules and any custom analytical models the customer might develop.

Data preparation for integrated analytics involves selecting data from the database for eligible resources. An eligible resource is any monitored resource that produces a meaningful result from the execution of one of the underlying analytical models (such as sensor and maintenance analytics).

Table 8-1 describes INTEGRATION_HEALTH_DATA_PREPARATION.str, the SPSS stream that is involved in preparing data for integration analytics.

Table 8-1 The INTEGRATION_HEALTH_DATA_PREPARATION stream

Stream name	INTEGRATION_HEALTH_DATA_PREPARATION.str
Purpose	This stream extracts the data from the RESOURCE_KPI and MASTER_PROFILE_VARIABLE tables, prepares the data for use by the model, and exports it to a comma-separated value (CSV) file.

Stream name	INTEGRATION_HEALTH_DATA_PREPARATION.str
Input	Input for this stream includes the health score information produced by the sensor and maintenance analytics processes, plus any available details about scheduled and forecasted maintenance activities. If the stream detects anomalies, it can also reanalyze some of the data used in the earlier analytical processes.
Output	<p>The stream output includes two CSV files:</p> <ul style="list-style-type: none"> ► Training_Eligibility_IntegratedAnalytics_Report.csv: This file includes a list of resources that are marked with either 1 (indicating there is sufficient data for analysis) or 0 (indicating there is not sufficient data for analysis). ► Integrated_health_data.csv: This file contains pivoted data from the underlying database tables, which forms the input for scoring (the next step in the training process). This file contains aggregated data that includes Sensor Health Score, Maintenance Health Score, Forecasted Days to Next Maintenance, and Scheduled Days to Next Maintenance. These parameters are aggregated for the assets found in the RESOURCE_KPI table.

The SPSS streams involved in the data preparation are shown in Figure 8-2, along with the output CSV file generation nodes.

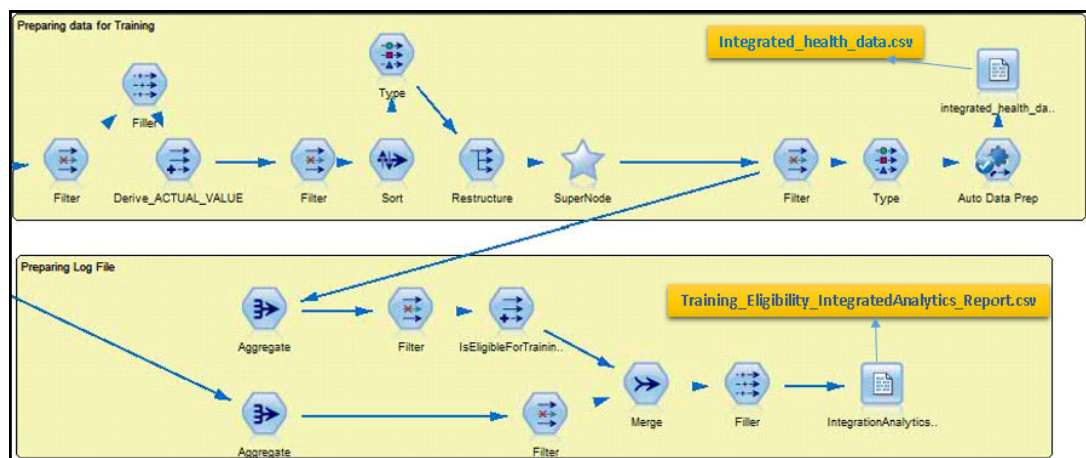


Figure 8-2 SPSS streams involved in data preparation

The INTEGRATION_HEALTH_DATA_PREPARATION stream is started by IBM Integration Bus.

8.4 Predictive modeling for integrated analytics

This section describes the workings of the integrated health score model from an SPSS standpoint, and also describes how IBM Integration Bus is used to start the model for scoring in real-time mode.

8.4.1 Integrated health score model

This section describes the SPSS model involved in the training and scoring part of the process (the integrated health score model), which computes the Integrated Health Score for each analyzed asset. Table 8-2 describes the stream responsible for this model, `INTEGRATION_HEALTH_COMBINED.str`.

`INTEGRATION_HEALTH_COMBINED.str` is based on the IBM SPSS Modeler auto classification model's confidence. The model provides options to modify the costs, revenue, and weights. The settings that you choose depend on the requirements of the data with which you are working. Depending on the data and requirements, you can use balancing to get better results.

Table 8-2 The `INTEGRATION_HEALTH_COMBINED` stream

Stream name	<code>INTEGRATION_HEALTH_COMBINED.str</code>
Purpose	This stream is used to train the models and refresh them for the scoring service
Input	The input data contains the sensor and maintenance health scores along with scheduled and forecasted maintenance details for the monitored resources.
Target	<code>IS_FAIL</code>
Output	The output of this stream is the Integrated Health Score of the monitored resource.

The Integrated Health Score for each resource is provided in a range from 0 to 1, with higher numbers indicating better asset health. If the input data model or structure is modified, the Integrated Health Score model must be retrained on the new data.

8.4.2 Real-time invocation of the model

After the integrated health score model is trained with historical data, it can be used to score newer data on a real-time basis. Scoring in real time is supported by an IBM Integration Bus (IIB)-based invocation of the appropriate SPSS stream. For scoring to happen in real time, the model is started by the PMQ solution's Service Adapter, which passes the input as a SOAP payload.

It is important to note that in PMQ, the sensor analytics model is triggered first, followed by the integration health score model in real-time mode. This is orchestrated by IIB. The integration health score model uses the real-time Sensor Health Score that is generated and the most recent Maintenance Health Score from the database.

Upon invocation by IIB, the integrated health score model performs the following profile calculations using the aggregated data from the `RESOURCE_PROFILE` table:

- ▶ Measurement of Type Count for measurement types that include:
 - Maintenance Health Score
 - Forecasted Days to Next Maintenance
 - Scheduled Days to Next Maintenance
- ▶ Measurement of Type for measurement types such as Health Score and Deviation.

The generated profiles are specified in the Service Adapter configuration using the respective profile codes for the invocation of the integrated health score model.

The Service Adapter configuration for the integrated health score model is shown in Figure 8-3.

o	orchestration	
o	orchestration_cd	pmq.measurement
o	step	
o	step	
o	step	
o	adapter_class	com.ibm.analytics.foundation.adapter.service.ServiceAdapter
o	adapter_configuration	
o	xs:type	ns3:service_adapter_configuration
o	service_invocation_configuration	
o	service_invocation_configuration	<service_invocation_configuration> <event_selector> <event_f
o	event_selector	
o	service_invocation_action	
o	profile_row_selector	
o	alias	MHS_PROFILE
o	shared_selector_cd	RESOURCE_PROFILE
o	key_field_value	
o	profile_field_name	PROFILE_VARIABLE_CD
o	value	MHS
o	key_field_value	
o	key_field_value	
o	key_field_value	
o	key_field_value	
o	profile_row_selector	
o	profile_row_selector	
o	profile_row_selector	
o	service_invocation	

Figure 8-3 Service Adapter configuration for the integrated health score model

The service invocation to trigger the integrated health score model results in the generation of yet another event in PMQ. This event is of the type RECOMMENDED and includes the Integrated Health Score and any associated recommendation. This event is processed using the standard Integration Bus message flows, with the orchestration engine running the steps previously defined for the RECOMMENDED event type.

The Profile Adapter orchestration definition XML is shown in Figure 8-4.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	recommended
orchestration_cd	pmq.recommended
event_orchestration_mapping	
orchestration	
orchestration_cd	pmq.recommended
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
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
value	IHS
profile_update_action	
observation_profile_update	
orchestration	

Figure 8-4 Profile Adapter configuration to process Integrated Health Score model results

The orchestration definition XML for Integrated Health Score events has two steps:

1. The Event Store Adapter persists Integrated Health Score events into the EVENT and EVENT_OBSERVATION tables.
2. The Profile Adapter performs a Measurement of Type calculation based on the current Integrated Health Score and stores the results in the RESOURCE_PROFILE table. The Profile Adapter configuration for this calculation is shown in Figure 8-4.

8.5 Deploying integrated health score models

The developed models are deployed and read by using IBM SPSS Deployment Manager and IBM SPSS Analytical Decision Management.

8.5.1 Deployment

Model deployment involves the use of configurable parameters that govern the behavior of the models at run time. Some of these parameters are used by downstream applications that are provided by the models while running the stream. If parameters are not provided, the models are designed to use the default values.

Figure 8-5 provides two deployment level parameters. Both parameters are related to training individual resources. The first parameter is a flag that indicates the need to train a resource and the second parameter identifies the resource that needs training.

Parameters	Deployment	Execution	Globals	Search	Comments	Annotations
Name	Long name	Storage	Value			
IS_1_RES_TRAIN	Resource Training required	Integer	0			
RESOURCE_ID	Resource identifier	Integer	595			

Figure 8-5 Stream parameters

The INTEGRATION_HEALTH_COMBINED stream helps in training the models and refreshes them for the scoring service. It uses auto modeling to identify the most suitable model for producing the wanted machine failure probability health score. The stream must refresh automatically each time, so the Model Refresh deployment option type is selected, as shown in Figure 8-6.

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:

INTEGRATED_...

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

Role

Data

Figure 8-6 Stream Deployment tab with Model Refresh option

After the INTEGRATION_HEALTH_ANALYTICS_JOB stream is started, the training and refresh of the models happens automatically. An INTEGRATED_HEALTH_SCORE service is configured and started by the Integration Bus flows, which helps in getting real-time scores for the asset.

As a first step of deploying the model, you must import IBMPMQ_INTEGRATED_ANALYTICS.pes (contains the streams and jobs that must be deployed) into the IBM SPSS C&DS Deployment Manager. The file contains the streams and jobs that are shown in Figure 8-7.

- IBMPMQ_INTEGRATED_ANALYTICS

IBMPMQ_INTEGRATED_ANALYTICS.str

INTEGRATION_FILE_STATUS_FAIL.str

INTEGRATION_FILE_STATUS_SUCCESS.str

INTEGRATION_HEALTH_ANALYTICS_JOB

INTEGRATION_HEALTH_COMBINED.str

INTEGRATION_HEALTH_DATA_PREPARATION.str

INTEGRATION_HEALTH_RECOMMENDATION.str

Figure 8-7 Content of IBMPMQ_INTEGRATED_ANALYTICS.pes

Table 8-3 provides details about the streams and jobs, along with their purpose in terms of integrated analytics.

Table 8-3 SPSS streams and jobs involved in Integrated Analytics

Streams and jobs	Description
INTEGRATED_HEALTH_DATA_PREPARATION.str	A data preparation stream that retrieves data from solution's tables and prepares it to be used in modeling. The eligible data is exported to a CSV file.
INTEGRATED_HEALTH_COMBINED.str	A combined stream that helps in training the models and refreshing them for the scoring service.
IBMPMQ_INTEGRATED_ANALYTICS	An SPSS C&DS job used to start the integrated health stream and the recommendation stream.
IBMPMQ_INTEGRATED_ANALYTICS.str	A stream that is auto-generated during model training and for real-time scoring.
INTEGRATED_RECOMMENDATION.str	An Analytical Decision Management stream used to give integrated health recommendations. For real-time scoring, an INTEGRATED_RECOMMENDATION service is created and started by Integration Bus flows.

In addition to these streams, the INTEGRATED_HEALTH_STATUS_FAIL and the INTEGRATED_HEALTH_STATUS_SUCCESS streams are used in the job to write the result status back into a file, which helps Integration Bus to orchestrate and start the subsequent flows.

Figure 8-8 depicts the complete integration analytics job flow.

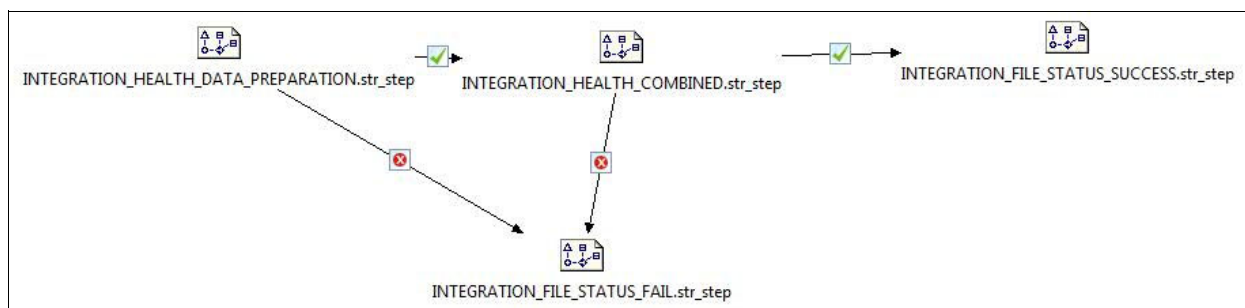


Figure 8-8 Integration analytics job flow

8.5.2 Recommendations from IBM SPSS Analytical Decision Management

IBM SPSS Analytical Decision Management brings the benefit of predictive analytics to real business problems, allowing you to build custom applications that are tailored to your customers or industry. Although applications are typically configured to solve specific problems, all are based on a common set of capabilities:

- ▶ Automate decisions using business rules.
- ▶ Add insight using predictive models.
- ▶ Use prioritization, optimization, or simulation to reach the best decision based on the above.

Depending on the health score calculated from the modeler, a recommendation is made regarding whether any urgent Inspection is needed (IHS101). Figure 8-9 shows some basic business rules created by using SPSS Decision Management.

	Rule name	Allocate to
1	Urgent Inspection IHS >= 0.7	IHS101
2	Need Monitoring IHS BETWEEN 0.4 and 0.7	IHS102
3	Remainder	IHS103

Figure 8-9 SPSS Decision Management rules

The recommendations are prioritized based on a set of parameters (Figure 8-10). The figure also displays the prioritization equation that determines how priorities are calculated. Prioritization depends on the probability of failure, the maintenance cost of an asset, its expected lifespan, and the downtime involved in maintenance. In general, the greater the risk, the higher it is prioritized. This determined risk is a simple calculation involving the impact of the failure of a critical asset and the propensity for it to fail during operation.

Prioritization Parameters

Max number of offers: 1

Campaign/Offer	Prob.to Fail	Maintenance Cost	Expected life time	Maintenance Downtime	Priority
Resource	HS		FDM		
IHS101	HS	10000	FDM	40	High
IHS102	HS	6000	FDM	40	Normal
IHS103	HS	3000	FDM	40	Low

Prioritization Model (Value to be maximized)

Prioritization Equation

(Prob.to Fail) (X Maintenance Cost) (-) (Expected life time) (X Maintenance Downtime)

Figure 8-10 Prioritization tab in SPSS Decision Management

Recommendations come in the form of codes such as IHS101, IHS102, and IHS 103, where IHS stands for Integrated Health Score and the numeric code denotes the recommendation as Urgent Inspection, Need Monitoring, or Need Monitoring within Limits, respectively.

This optimized recommendation provided from IBM SPSS Analytical Decision Management is read in real time using a web services invocation through Integration Bus message flows.

8.6 Integrated Analytics Report

The Maintenance Overview Dashboard (Figure 8-11) provides insights based on existing maintenance data and can include sensor data as your organization's data matures. The report shows the Sensor Health Score, Maintenance Health Score, and the Integrated Health Score for each location and resource.



Figure 8-11 Maintenance Overview Dashboard



TopN Failure analytics

Identifying the leading factors that cause manufacturing assets to fail is important in reducing long-term maintenance costs. However, responding to and repairing failures is just one part of the picture. Even greater efficiencies can be obtained if you can identify what is causing the failures so you can predict when they will occur in the future.

This chapter describes how the Predictive Maintenance and Quality (PMQ) solution enables top failure predictor analytics, or TopN Failure analytics.

The chapter includes the following sections:

- ▶ Business understanding of TopN Failure analytics
- ▶ Data understanding of TopN Failure analytics
- ▶ Data preparation for TopN Failure analytics
- ▶ Predictive modeling for TopN Failure analytics
- ▶ TopN Failure model invocation
- ▶ TopN Failure model deployment
- ▶ TopN Failure Analysis Report

9.1 Business understanding of TopN Failure analytics

Top failure predictors are a set of factors considered likely to cause failure of a particular manufacturing asset. The factors vary depending on the enterprise and business process.

The objective of TopN Failure analytics is to identify the factors with the strongest correlation to failures. The identified factors are then ranked in order of importance to enable complete analysis of any single one of them.

TopN Failure analytics examines resources that are known to have experienced frequent failure episodes.

9.2 Data understanding of TopN Failure analytics

TopN Failure analytics is implemented on sensor data that is stored in the solution's EVENT table and several master tables such as MASTER_MEASUREMENT_TYPE and MASTER_PROFILE_VARIABLE. In addition, the RESOURCE_KPI table is used to hold aggregated daily sensor data for the monitored resources, which is used in preparation for model training and scoring.

Various methods are available to understand the data better, including the Data Audit window in IBM SPSS Modeler (see Figure 9-1), which displays summary statistics, histograms, and distribution graphs about the listed data.

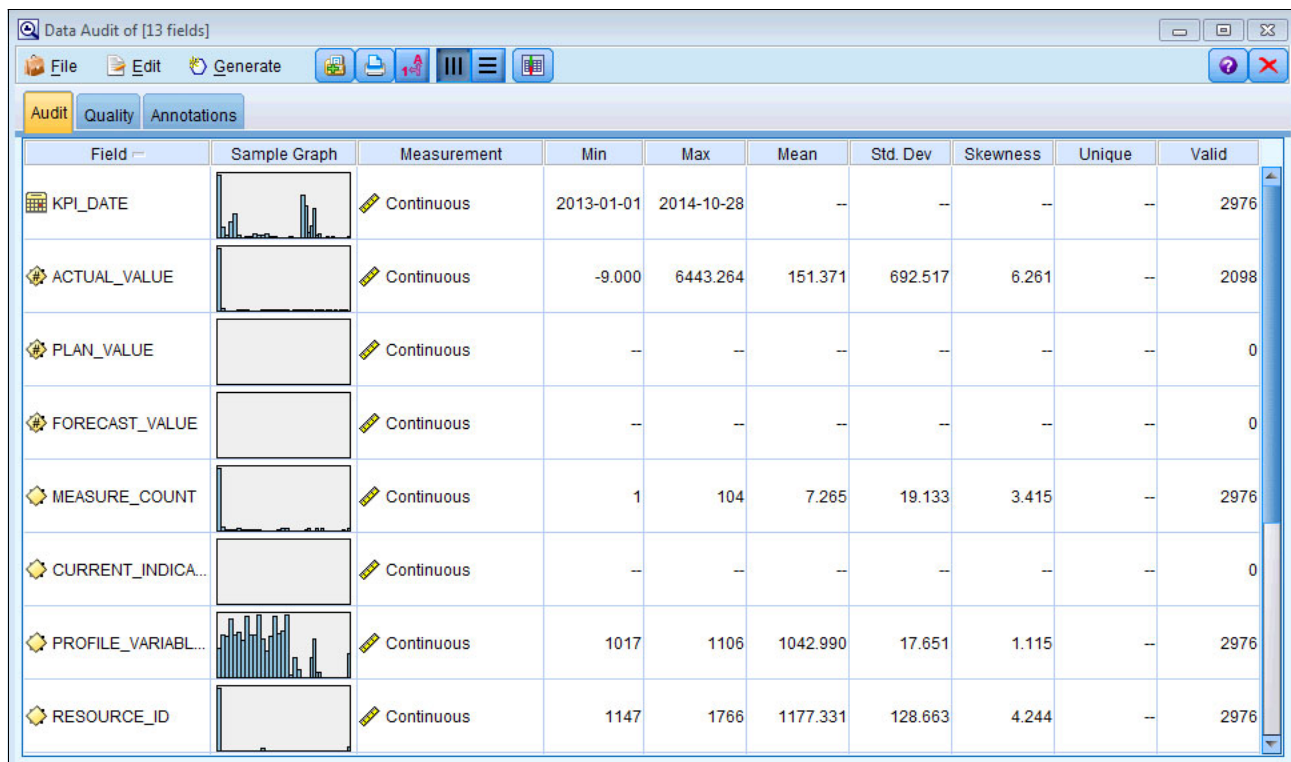


Figure 9-1 Using SPSS Modeler to learn more about TopN Failure data

9.3 Data preparation for TopN Failure analytics

Sensor data is processed and loaded into the PMQ data store using the solution's event processing flow (see Chapter 5, "Event mapping and processing" on page 61).

Depending on the data and the company's analytical goals, different kinds of data preparation can occur:

- ▶ Merging data sets and records (such as master data and event data)
- ▶ Selecting sample subsets of data to study specific resources and profiles
- ▶ Adding new fields associated with profiles created from sensor data
- ▶ Removing fields that are not required for TopN Failure analysis

The same basic rule applies to data preparation for TopN Failure analytics as applies to all analytical efforts: Sufficient data is needed to confidently identify the targeted patterns. So for optimal results, the data store needs to have a pre-determined number of days' worth of sensor data for each resource and parameter being studied. Generally speaking, larger data sets result in more accurate predictions.

9.4 Predictive modeling for TopN Failure analytics

The TOPN_MODEL stream is used to identify and rank the parameters that are most influential in predicting failure of an asset. This stream includes logistic regression (which measures the relationship between the categorical dependent variable and one or more independent variables) to build a FAIL model that identifies the scores for each Measurement Type identified as a one of the top factors that cause failure.

The stream uses the scripting capability of SPSS Modeler. The Execution tab (Figure 9-2) shows the sequence of steps the stream runs to produce the TopN.xml file using the highlighted path.

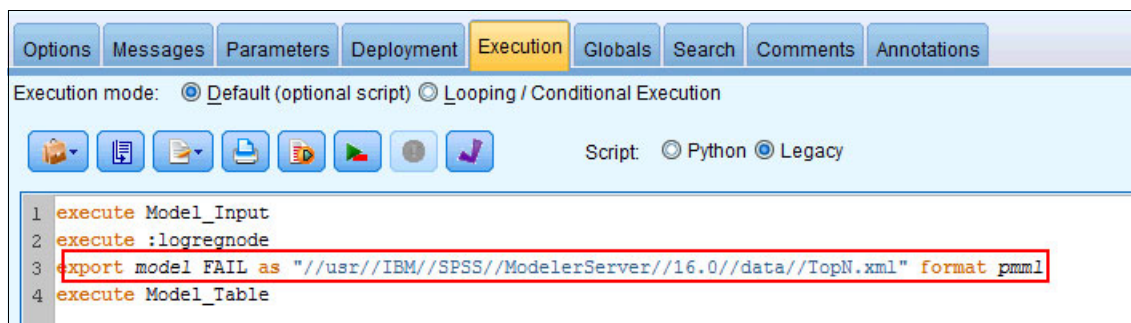


Figure 9-2 Execution tab showing script and TopN.xml path

To obtain the appropriate predictive importance of each profile (as calculated by sensor analytics processes), the model FAIL is exported in the Predictive Model Markup Language (PMML) format that is required by the TOPN_XML stream, as shown in Figure 9-3.

XML_data source: /usr/IBM/SPSS/ModelerServer/16.0/data/TopN.xml

XML schema: MiningSchema

Records (XPath expression): /PMML/GeneralRegressionModel/MiningSchema/MiningField

☒ Read all data ☐ Specify data to read

Fields: XPath Location

Figure 9-3 Importing PMML using XML Source Node

The XML source is then merged with the data in the RESOURCE_KPI table to determine the top predictors for each monitored resource. As shown in Figure 9-4, the failure importance of each profile, along with their cumulative values, are calculated and stored in a table for use in reports.

PARAMETER	IMPORTANCE	CUMULATIVE_IMPORTANCE
REPC	0.279	0.279
LUBE	0.112	0.391
RELH	0.096	0.486
OPHR	0.090	0.576
INSP	0.090	0.666
RPM	0.086	0.753
R_F1	0.084	0.837
R_B1	0.082	0.919
TEMP	0.081	1.000

Figure 9-4 List of predictors and their importance values

9.5 TopN Failure model invocation

The PMQ solution includes IBM Integration Bus message flows to start TopN Failure analytics and process the results. Processing occurs as shown in Figure 9-5 (the flow-dependent steps are shown in yellow boxes).

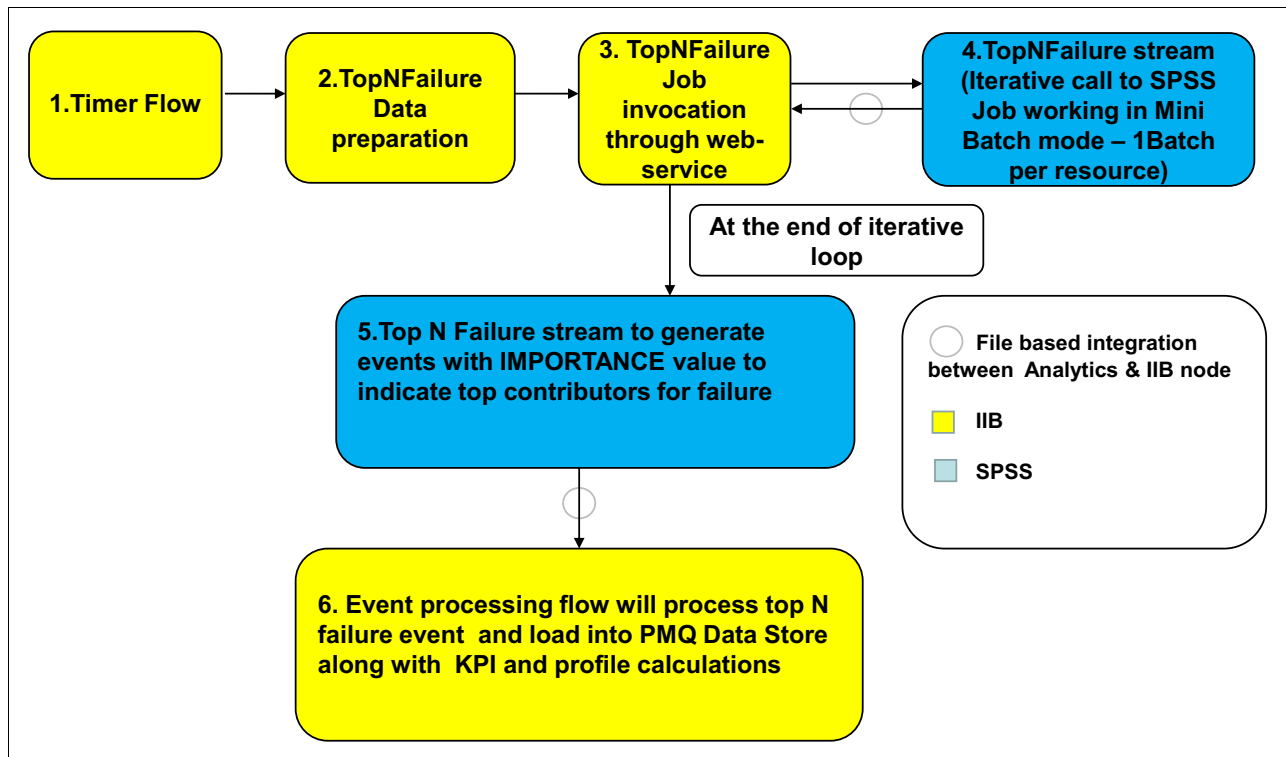


Figure 9-5 Processing flow for TopN Failure analytics

The next subsections provide details about each of these message flows that drive TopN Failure analytics.

9.5.1 TopN Failure Timer

The TopN Failure Timer message flow (refer to Figure 9-6) runs at a pre-scheduled time each day to initiate TopN Failure analytics. The compute node prepares a TimeoutRequest message to trigger the TopN Failure data preparation flow.

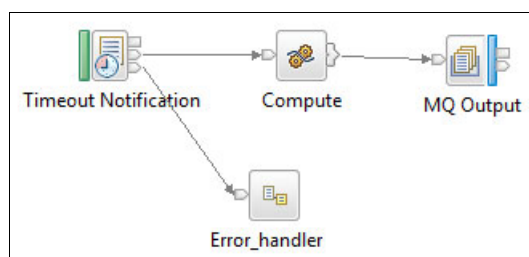


Figure 9-6 TopN Failure Timer message flow

9.5.2 TopN Failure Data Preparation

The TopN Failure Data Preparation flow (see Figure 9-7) prepares the list of resources that are eligible for TopN Failure analytics. The list is then passed to the TopN Failure Job Integration flow. This message flow also clears the table, TOPN_PREDICTORS, in which SPSS stores the results of TopN Failure analytics.

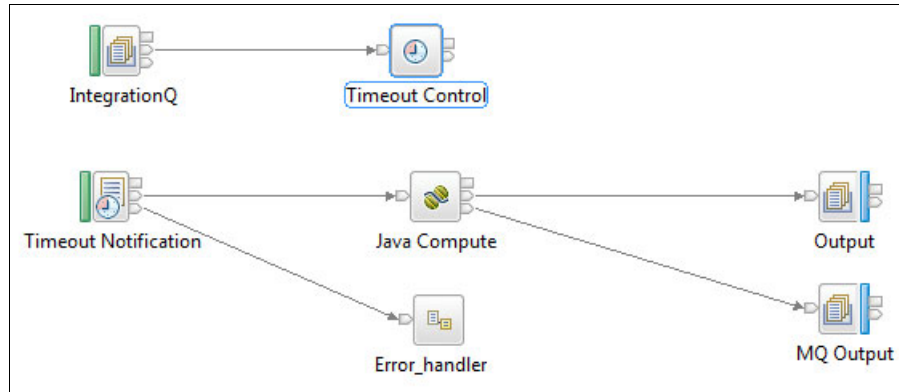


Figure 9-7 TopN Failure Data Preparation message flow

9.5.3 TopN Failure Job Integration

The TopN Failure Job Integration message flow (see Figure 9-8) starts the SPSS IBMPMQ_TOP_FAILURE_PREDICTORS_JOB to process the list of resources loaded in the RESOURCE_KPI table. The job is run for each resource and then sends status updates to the Integration Bus node. The flow then starts the job for another resource and this process continues until the complete list of resources is processed. The flow then starts the SPSS IBMPMQ_TOP_FAILURE_EVENTS_JOB, which generates SPSS results from the TOPN_PREDICTORS table, converts them to PMQ events, and places them in a shared directory (/integrationin) on the Integration Bus node. TopN Failure analytics results are referred to as the TopN recommendation events. These events are processed by the solution's regular event processing flow.

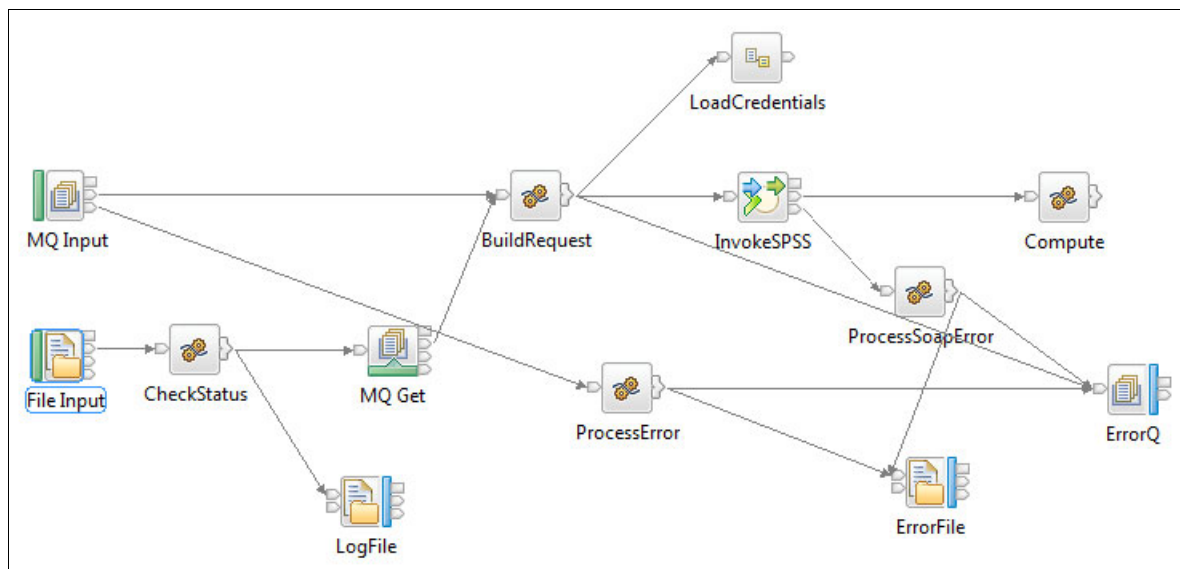


Figure 9-8 TopN Failure Job Integration message flow

9.5.4 TopN Failure Analytics Result Processing

As was explained, the TopN Failure event generation stream generates results that are converted to TopN recommendation events. These events contain the TopN Failure analytics results with an importance value for each parameter that was configured for TopN Failure analytics (see Figure 9-4 on page 128). The solution's event processing flow processes these events according to the TopN Failure orchestration definition XML shown in Figure 9-9.

?? xml	version="1.0" encoding="UTF-8"
e ns3:orchestration_definition	
a xmlns:Q1	http://www.ibm.com/asf
a xmlns:ns3	http://www.ibm.com/asf/orchestration
a xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
a xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
e event_orchestration_mapping	
e event_orchestration_key_cd	topnrc
e orchestration_cd	pmq.topnrc
e orchestration	
e orchestration_cd	pmq.topnrc
e step	
e adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
e adapter_configuration	

Figure 9-9 TopN Failure orchestration definition XML

The TopN Failure orchestration definition XML has only one step:

- The Profile Adapter performs calculations on the TopN recommendation events according to the Profile Adapter configuration that is defined in the TopN Failure orchestration definition XML. The calculations involve aggregating the data carried by the events. The aggregated data is then persisted into the RESOURCE_PROFILE table.

The Profile Adapter configuration for one measurement type, TEMP_IMPORTANCE, is shown in Figure 9-10.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	topnrc
orchestration_cd	pmq.topnrc
orchestration	
orchestration_cd	pmq.topnrc
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xsitype	ns3:profile_adapter_configuration
observation_profile_update	
observation_selector	
profile_update_action	
profile_row_selector	
shared_selector_cd	RESOURCE_PROFILE
key_field_value	
profile_field_name	PROFILE_VARIABLE_CD
value	TEMP_IMPORTANCE
shared_calculation_invocation_group_cd	PROFILE_COUNT
observation_profile_update	
observation_profile_update	
observation_profile_update	
observation_profile_update	
observation_profile_update	
observation_profile_update	
observation_profile_update	
observation_profile_update	

Figure 9-10 Profile Adapter step for TopN recommendation events

9.6 TopN Failure model deployment

All SPSS Modeler streams are deployed into the SPSS Collaboration and Deployment Services (C&DS) Deployment Manager, where a single IBMPMQ_TOP_FAILURE_PREDICTORS.pes file holds all of the content, as shown in Figure 9-11.

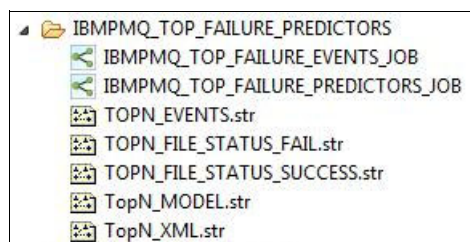


Figure 9-11 Contents of IBMPMQ_TOP_FAILURE_PREDICTORS.pes file

A folder called IBMPMQ_TOP_FAILURE_PREDICTORS is created. The folder contents are described in the Table 9-1, along with their purpose in TopN Failure analytics.

Table 9-1 TopN Failure streams and jobs

Streams and jobs	Description
TOPN_MODEL.str	This stream uses a logistic regression model to identify the scores for each of the measurement types (parameters) that include the top failure-related factors. The output is a PMML file containing the predictive importance of each profile.
TOPN_XML.str	This stream is a data preparation stream that reads the PMML file, merges it with the KPI table, and then performs computations. The output is sorted data reflecting the cumulative importance of the identified failure factors.
IBMPMQ_TOP_FAILURE_PREDICTORS_JOB	This is an SPSS C&DS job that is used to start the TopN Failure-related streams with success or failure status.
TOPN_EVENTS.str	This stream creates a CSV file containing the top factors and profiles affecting failure events.
IBMPMQ_TOP_FAILURE_EVENTS_JOB	This is an SPSS C&DS job that starts the TopN_Events stream.

9.7 TopN Failure Analysis Report

The TopN Failure Analysis Report (see Figure 9-12) shows the factors (in the form of profile variables) that contribute to the failure of a resource. Each profile has an importance value that is expressed as a percentage.

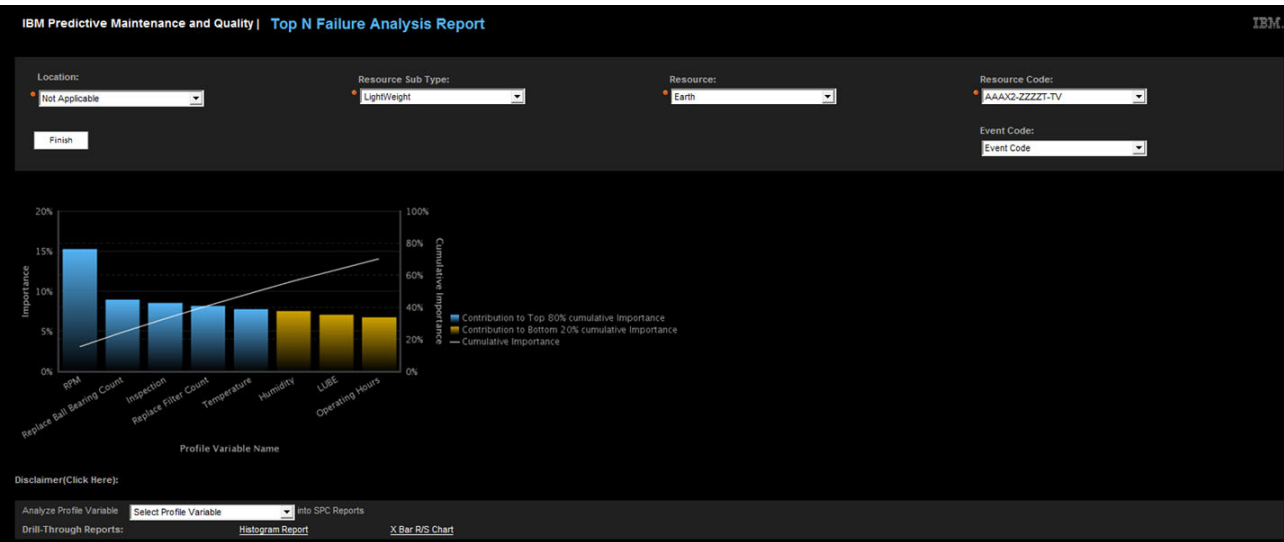


Figure 9-12 TopN Failure Analysis Report

The profile is indicated on 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 specific resource. The profile values that are displayed in blue contribute the most (top 80%) to the resource failure. The profile importance values that display in yellow make up the bottom 20% of the factors that cause failure. The curved line on the chart indicates the cumulative importance value. Drill through hyperlinks to access statistical process control reports are provided at bottom of the report page.



Statistical process control support

This chapter describes the Predictive Maintenance and Quality solution's support for statistical process control (SPC), including the built-in control charts and histograms that help you monitor the stability of processes and track variations in critical operating parameters.

Using a set of connected dashboards, you can perform root cause analysis on quality issues by combining standard PMQ reports (specific to the solution's various analytical models) with SPC-specific charts. This advanced integration brings a more holistic perspective to tracking process variability.

This chapter includes the following sections:

- ▶ Histogram Report
- ▶ SPC - X Bar R/S Chart
- ▶ Integrating histogram and SPC charts for advanced analysis

10.1 Histogram Report

The Histogram Report (Figure 10-1) shows the frequency distribution of a particular parameter of a resource (such as a machine or tool), which helps identify any gaps between the expected value and the actual one. The Y Axis shows the frequency. The X Axis shows the bins. The height of the bar in a bin indicates how often the event falls into the specified range. In this view, most of the customizable parameters (number of bins, bin width, and so on) are automatically computed and adjusted to produce an optimal initial view of the underlying data.

The report includes an option to customize the number of bins and bin widths to override the default values. Extra options are also provided to remove the frequency distribution from either or both of the ends of the range as a way of focusing on a different range than the default one.



Figure 10-1 Histogram Report

The histogram plot includes an additional descriptive statistics table that displays central tendency measures. These measures include Mean, Median, Variation Measure (for example, Range), Min, Max, Standard Deviation, and Distribution Statistics (including Skewness and Kurtosis). Links to other SPC charts, such as the X Bar - R/S Chart, are also provided. This makes it easier to establish an analytical trial.

The filters applied to the Histogram Report are Date, Location, Resource Sub Type, Resource, Resource Code, Measurement Type, Event Type, and Event Code. Extra filter options are provided so you can customize the default bin range, number of bins, and the minimum and maximum data set ranges.

10.2 SPC - X Bar R/S Chart

The SPC - X Bar R/S Chart is essentially a dashboard that includes two subordinate charts: An X Bar chart and, depending on the situation, either a range (R) chart or a standard deviation (S) chart. This information helps in applying SPC techniques to monitored processes. You can visualize the variance and mean plots, identify threshold breaches, and spot the emergence of patterns that indicate a process that is not working properly.

Users can select the best chart combination for their needs, such as showing the X Bar chart in the left pane and an R or S chart in the right pane for analysis of selected subgroups. Each chart shows the data points in relation to the mean value, and upper and lower control limits. In addition, filters can be applied such as Date, Location, Resource Sub Type, Resource, Resource Code, Event Type, Measurement Type, and Event Code.

10.2.1 X Bar and R charts

The X Bar and R charts are the pair of control charts used to monitor the process behaviors and results over time based on samples collected at regular intervals. These charts help determine whether a process is stable and predictable.

The X Bar chart shows how the mean changes over time, whereas the R chart shows how the range of the subgroups changes over time. For the R chart, the sample size is typically small (10 or less). Figure 10-2 shows the X Bar and R chart combination.

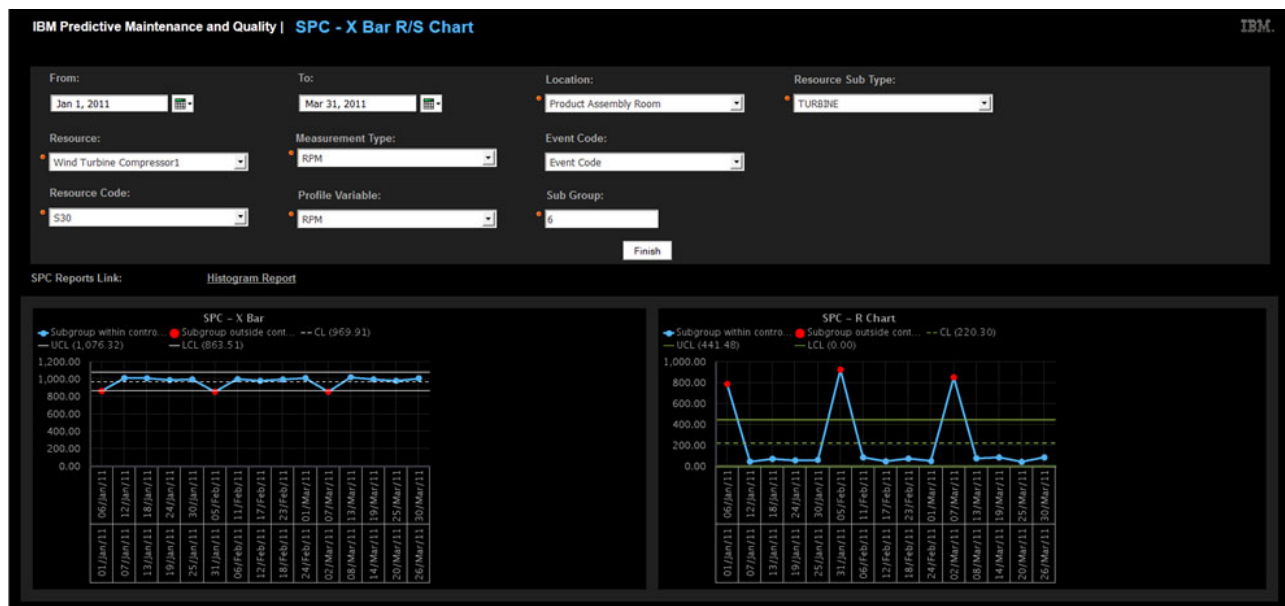


Figure 10-2 SPC - X Bar R/S Chart with R Chart shown in lower right corner

10.2.2 X Bar and S charts

The X Bar and S chart combination is used when the size of the subgroups is large. The R chart is considered a poor statistic from which to estimate subgroup distributions, whereas the S chart is better for this purpose. In this case, the X Bar chart displays control limits that are calculated by using the average standard deviation. The S Chart shows the standard deviation of the subgroups.

Figure 10-3 shows the X Bar and S chart combination. The solution automatically substitutes the R chart with the S chart when the user selects a sample size value greater than 10.

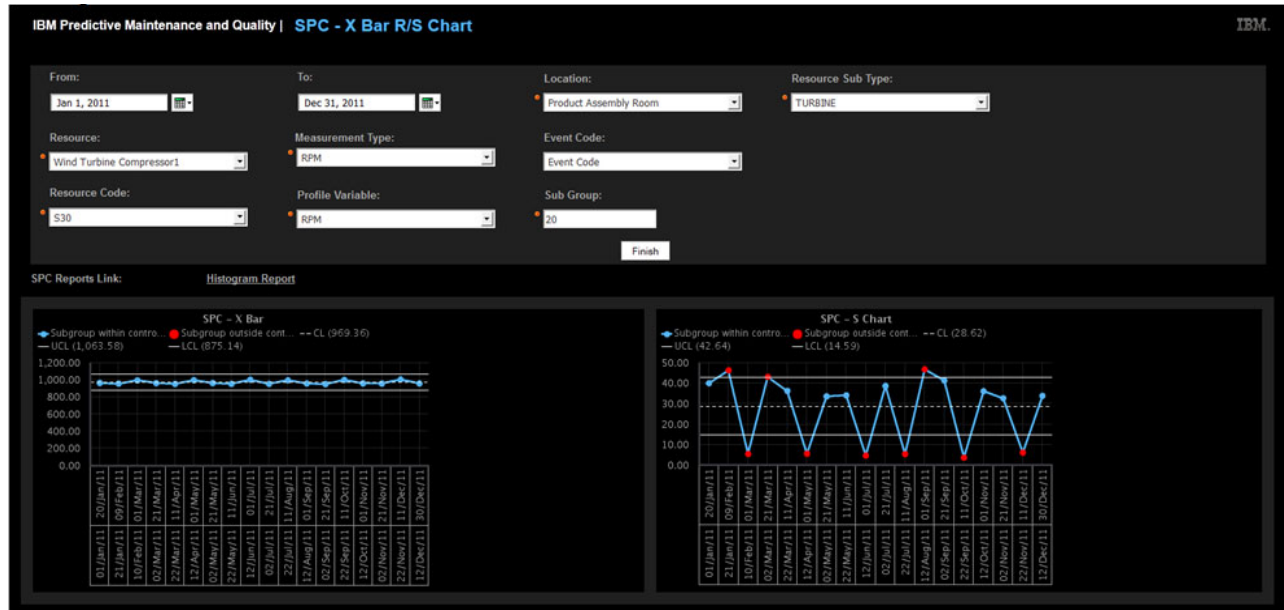


Figure 10-3 SPC - X Bar R/S Chart with S chart shown in lower right corner

10.2.3 Customizing the SPC - X Bar R/S Chart

Based on the subgroup value entered by the user, the R Chart or the S Chart is rendered along with X Bar chart. For example, if the subgroup value is 10 or less, the R chart is rendered. If it is greater than 10, the S Chart is displayed.

The SPC - X Bar R/S Chart requires the user to select values. Table 10-1 lists the prompts for these values.

Table 10-1 Prompts for value selection

Prompt	Purpose
From Date	Starting date for sample analysis
To Date	End date for sample analysis
Location	The location name, if needed for analysis
Resource Sub Type	The resource sub type name
Resource	The resource name
Resource Code	The resource code
Measurement Type	The measurement type
Profile Variable	The profile variable
Sub Group	The subgroup to be displayed
Event Code	An optional prompt allowing selection of event codes

If a sample falls above or below the designated values, then the data points are shown on the application window in RED, indicating that the process has gone out of control and needs immediate action.

Even if the plotted points have not breached the threshold, you might be able to discern patterns that indicate atypical variations in process performance. As you gain experience with the solution, identifying these patterns becomes easier.

10.3 Integrating histogram and SPC charts for advanced analysis

Figure 10-4 shows the TopN Failure Analytics report to provide an example of the option to drill through to the applicable histograms and X Bar R and S charts.

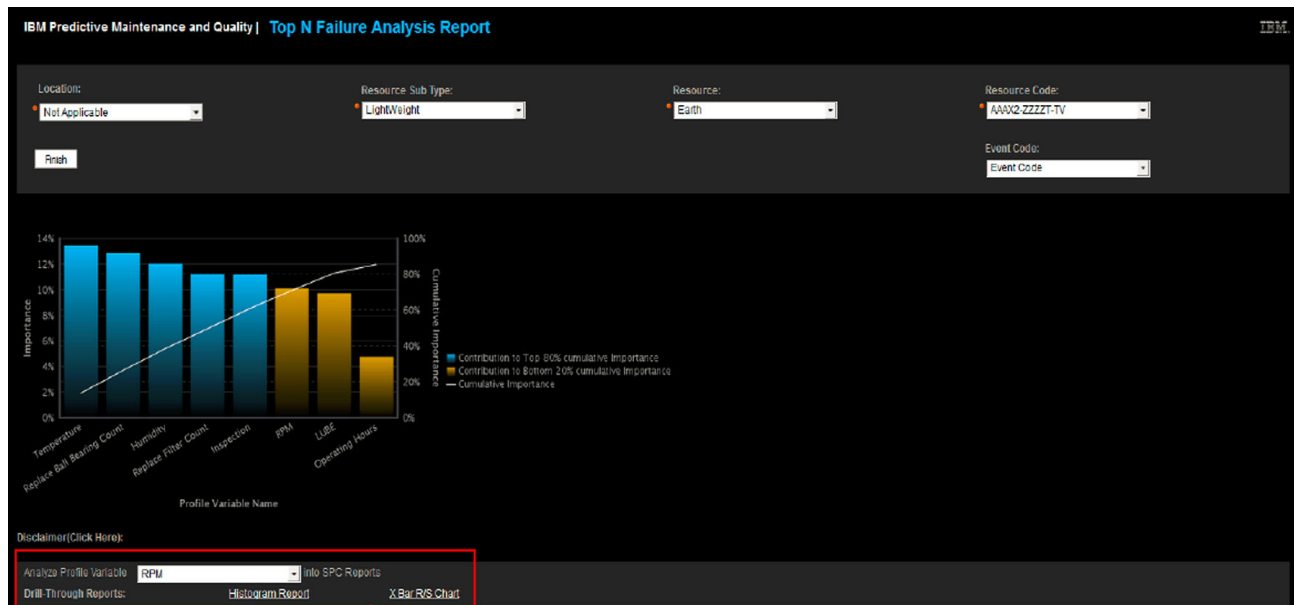


Figure 10-4 TopN Failure report with drill through option

Users can move between the TopN Failure report and the associated histogram reports and X Bar R and S charts to analyze the data in depth. The Histogram report can help identify gaps between processes, whereas the X Bar R and S charts can help monitor process behavior and performance.



Inspection and warranty analytics

This chapter explains how inspection and warranty analytics are performed in the IBM Predictive Maintenance and Quality solution using the Quality Early Warning System algorithm.

This chapter includes the following sections:

- ▶ Business scenarios
- ▶ Inspection analytics
- ▶ Warranty analytics

11.1 Business scenarios

The Quality Early Warning System (QEWS) in the IBM Predictive Maintenance and Quality solution detects emerging quality problems sooner and with fewer false alarms than is typical with traditional statistical process control. To achieve this improvement, QEWS is sensitive to subtle changes in data values, such as shifts that are small in magnitude or trends that grow slowly over time. This sensitivity means that QEWS usually needs fewer data points than more traditional methods to obtain the same level of statistical confidence.

Early detection of quality problems can help companies avoid various negative outcomes:

- ▶ Large inventories of unsold, defective products, resulting in high scrap costs
- ▶ Shipments of defective products, resulting in high warranty expenses
- ▶ Reliability problems in the field, resulting in damage to brand value
- ▶ Delayed order fulfillments thanks to compromised production of supply-constrained materials

The subjects of QEWS analysis are *products*. A product is typically a part or a part assembly, but it can also be a process or a material. Products are often used in larger finished assemblies, which in QEWS are referred to as *resources*.

QEWS has two primary use cases:

- ▶ Quality inspections
- ▶ Warranty issues

11.1.1 Quality inspections

Defects occur in a manufacturing setting due to variations in factors such as process, raw materials, design, and technology. The resulting low product quality creates an ever-larger inventory of defective lots and increased need for inspection. These defects also cause increases in failure rates.

QEWS evaluates evidence to determine whether the rate of defects and failures meets the company's standards for what is acceptable. QEWS highlights situations for which the evidence exceeds a specified threshold. QEWS can detect emerging trends earlier than traditional statistical process control. After QEWS issues a warning, analysis of the resulting charts and tables can identify the problem's point of origin, nature, and severity, along with the current state of the associated process.

The QEWS inspection algorithm analyzes data from the inspection, testing, or measurement of a product or process operation over time. The data can be obtained from these 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, results of satisfaction surveys)

Based on your needs, you can adjust the frequency at which data is captured and input into QEWS, and the frequency at which QEWS is run. For example, monitoring the quality of assemblies procured from a supplier might be done weekly, but monitoring units moving through an active manufacturing line might better be done daily.

11.1.2 Warranty issues

The warranty analytics algorithm can be used by a manufacturing company in several ways. Its primary purpose lies in identifying conditions that lead to accelerated wear and warranty-based replacement of parts. Additional uses involve helping companies determine appropriate warranty pricing (including extended warranties) and identifying vendors.

The PMQ solution's Warranty Analytics feature is an implementation of an advanced version of QEWS called the Quality Early Warning System for Lifetime (QEWSL) applications algorithm. The core QEWSL algorithm is applied to three specific use cases that involve sales, production, and manufacturing, each of which differs in terms of data loading and data transformation.

Table 11-1 provides brief details about the use cases.

Table 11-1 Use cases to which the QEWS algorithm is applied

Use case	Details
Sales	<p>This use case identifies variations in wear and replacement rates as they are being aggregated, based on sales date. The date of sale can be a key factor in certain warranty situations.</p> <p>For example, if parts carrying a common one-year warranty are suffering failure in cold climates at a specific location, it is possible that the parts installed during winter seasons experience rapid wear during their initial service months, followed by delayed wear through the end of their warranty period. These variations affect the parts' wear rates and weighted replacement rates, which can be important to the operation of the QEWSL algorithm.</p>
Production	<p>This use case identifies variations in wear and replacement rates of a specific part within a resource, as aggregated since the day the part was produced. The production date can have correlations to part quality or process-related issues.</p> <p>For example, if a batch of parts produced on a given date experiences rapid wear that leads to replacements, the cause might be traced to a manufacturing condition or some production variable that was present on that date.</p>
Manufacturing	<p>This use case identifies the variation in wear and replacement rates of a specific part type within a resource, as aggregated since the day the part was produced.</p> <p>For example, if particular batches of resources have certain parts that are failing prematurely due to process issues with the associated lot or assembly line, this indicates that there are issues with that lot or line.</p>

11.2 Inspection analytics

The Predictive Maintenance and Quality solution uses the QEWS algorithm to inspect data and detect unfavorable changes in the quality of products by plotting the QEWS-generated results in failure rate and evidence charts. The QEWS algorithm needs batch (aggregated) data whereas the PMQ solution supports two modes of data loading for inspection:

- ▶ Batch mode
- ▶ Real-time mode

Processing in both of these modes is illustrated in Figure 11-1. The QEWS algorithm is started in batch mode flow.

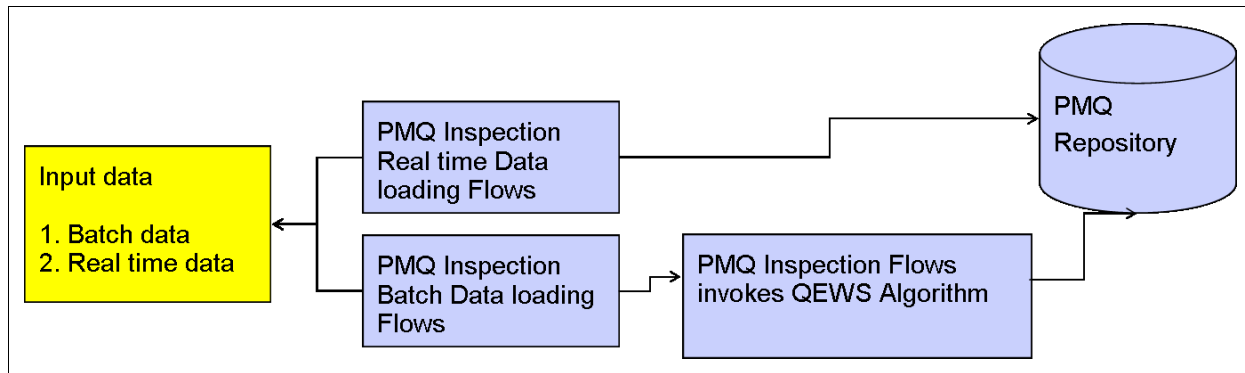


Figure 11-1 Inspection Data Processing

11.2.1 Input data

The PMQ solution needs master data such as Product (Part), Product Parameter, and Production Batch to process inspection data. This data is loaded by using PMQ master data flows.

The QEWS algorithm needs these product parameters to process inspection data:

- ▶ LAM0: Acceptable fallout rate
- ▶ LAM1: Unacceptable fallout rate
- ▶ PROB0: Probability of not having a false alarm
- ▶ INSPECT_NO_DAYS: Number of days for which the QEWS algorithm will process data

Inspection data can be processed in two modes:

- ▶ Batch mode: In this mode, a batch data file is placed in the /batchdata in folder of the ESB node. The batch data file should contain data for historical dates.
- ▶ Real-time mode: In this mode, a scheduler configured in IBM Integration Bus is triggered once each day to process both historical and real-time data.

11.2.2 Inspection batch data loading message flow

The QEWS algorithm needs inspection data aggregated by day for each product. Integration Bus message flows accept batch data in form of comma-separated values (CSV) files. The batch data is used primarily to load historical data for the product. The batch data for one product is as shown in Example 11-1.

Example 11-1 Batch data for a product

product_cd	product_type_cd	inspection_date	qty_produced	inspected	failed	language_cd	tenant_cd
PPR-00000001	PPA-00000003	2014-01-03	1904	1904	4	EN	PMQ
PPR-00000001	PPA-00000003	2014-01-04	3155	3155	15	EN	PMQ
PPR-00000001	PPA-00000003	2014-01-05	0815	0815	3	EN	PMQ
PPR-00000001	PPA-00000003	2014-01-06	1450	1450	7	EN	PMQ
PPR-00000001	PPA-00000003	2014-01-07	2554	2554	19	EN	PMQ
PPR-00000001	PPA-00000003	2014-01-08	4651	4651	28	EN	PMQ

ProcessBatchData message flow

The message flow shown in Figure 11-2 processes the inspection data. The input file contains data for each product and product type combination that is aggregated by day. Each row in the file reports data points such as quantity produced, number tested, and number failed. The QEWS inspection algorithm needs the file content and the parameters associated with the product and product type combination. This message flow parses the CSV file and updates the PRODUCT_KPI table with the raw values reported in the file. After the incoming data is persisted in the PRODUCT_KPI table, the QEWS algorithm is started by the flow through an IBM WebSphere MQ message trigger.

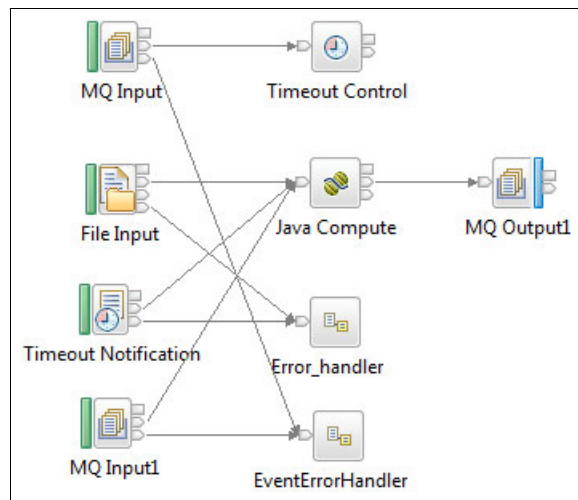


Figure 11-2 ProcessBatchData Message Flow

The ProcessBatchData message flow is also triggered when a timeout request message is sent to the WebSphere MQ queue. Upon receipt of the timeout request, the message flow prepares data for the QEWS algorithm, including both real-time and historical data. After the data is prepared, the QEWS algorithm is started for all of the products that are loaded in the PRODUCT_KPI tables

The results generated by the QEWS algorithm are persisted into the PRODUCT_KPI and PRODUCT_PROFILE tables, and the algorithm-generated charts are stored on the Integration Bus node. IBM Cognos Business Intelligence picks up these charts from the

Integration Bus node through network file sharing. The location on the server where QEWS charts are saved is configured in the `loc.properties` file stored within the Integration Bus shared classes.

11.2.3 PMQ Inspection Timer message flow

The Inspection Timer message flow (Figure 11-3) is scheduled to run once each day. It puts a message in the WebSphere MQ message queue to trigger the `ProcessBatchData` message flow.

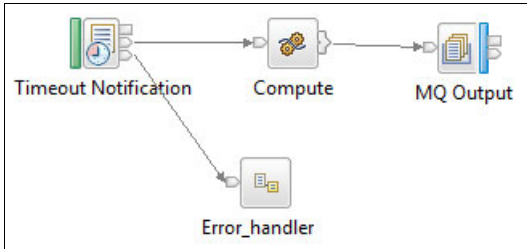


Figure 11-3 Inspection Timer message flow

11.2.4 PMQ inspection real-time data loading flow

Inspection-related real-time data is processed as events in the PMQ solution using the standard event processing flows explained in the Chapter 5, “Event mapping and processing” on page 61. Two types of events are reported for inspection data:

- ▶ Events with the event type Production
- ▶ Events with the event type Inspection

The events are processed by using the orchestration rules defined for each respective event type.

Production events

Production events report the quantity of an item produced and are processed by using the inspection orchestration XML shown in Figure 11-4.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	production
orchestration_cd	pmq.production
orchestration	
orchestration	
orchestration_cd	pmq.production
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	

Figure 11-4 Production orchestration XML

Production events have only one orchestration step:

- Profile Adapter performs calculations on the event data using the profile adapter configuration defined in the inspection orchestration XML as shown in Figure 11-5. For production events, a Measurement of Type Count calculation is performed and the results are persisted in the PRODUCT_KPI table.

?? xml	version="1.0" encoding="UTF-8"
e ns3:orchestration_definition	
a xmlns:Q1	http://www.ibm.com/asf
a xmlns:ns3	http://www.ibm.com/asf/orchestration
a xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
a xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
e event_orchestration_mapping	
e event_orchestration_mapping	
e event_orchestration_key_cd	production
e orchestration_cd	pmq.production
e orchestration	
e orchestration	
e orchestration_cd	pmq.production
e step	
e adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
e adapter_configuration	
a xsi:type	ns3:profile_adapter_configuration
e observation_profile_update	
e observation_selector	
a table_cd	EVENT_OBSERVATION
e observation_field_value	
e field_name	MEASUREMENT_TYPE_CD
e value	QTY
e profile_update_action	
e profile_row_selector	
e shared_selector_cd	PRODUCT_KPI
e shared_calculation_invocation_group_cd	PRODUCT_KPI_QTY_COUNT

Figure 11-5 Profile adapter step for Production events

Inspection events

The events of type Inspection carry two kinds of measurement types:

- no_tested that represents the number of times a product is tested for defects or failure
- no_failed that represents the number of times the tests failed, indicating a problem

These events are processed using the inspection orchestration XML shown in Figure 11-6.

?? xml	version="1.0" encoding="UTF-8"
e ns3:orchestration_definition	
a xmlns:Q1	http://www.ibm.com/asf
a xmlns:ns3	http://www.ibm.com/asf/orchestration
a xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
a xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
e event_orchestration_mapping	
e event_orchestration_key_cd	inspection
e orchestration_cd	pmq.inspection
e event_orchestration_mapping	
e orchestration	
e orchestration_cd	pmq.inspection
e step	
e adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
e adapter_configuration	
e orchestration	

Figure 11-6 Inspection orchestration XML

Inspection events have only one orchestration step:

- ▶ Profile Adapter performs calculations on the event data using the profile adapter configuration defined in the inspection orchestration XML as shown in Figure 11-7. For inspection events, a Measurement of Type Count calculation is performed and the results are persisted into the PRODUCT_KPI table.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	inspection
orchestration_cd	pmq.inspection
event_orchestration_mapping	
orchestration	
orchestration_cd	pmq.inspection
step	
adapter_class	com.ibm.analytics.foundation.adapter.profile.ProfileAdapter
adapter_configuration	
xs:type	ns3:profile_adapter_configuration
observation_profile_update	
observation_selector	
table_cd	EVENT_OBSERVATION
observation_field_value	
field_name	MEASUREMENT_TYPE_CD
value	INSPECT
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	PRODUCT_KPI_INSPECT_COUNT
observation_profile_update	
observation_selector	
table_cd	EVENT_OBSERVATION
observation_field_value	
field_name	MEASUREMENT_TYPE_CD
value	FAIL
profile_update_action	
profile_row_selector	
shared_calculation_invocation_group_cd	PRODUCT_KPI_FAIL_COUNT
orchestration	

Figure 11-7 Profile adapter step for Inspection events

11.2.5 Inspection Chart

Starting the QEWS Inspection algorithm generates charts that are stored onto the file system of the ESB node. These charts are pulled through network file sharing and displayed in Cognos Business Intelligence reports.

The QEWS Inspection Chart is shown in Figure 11-8.

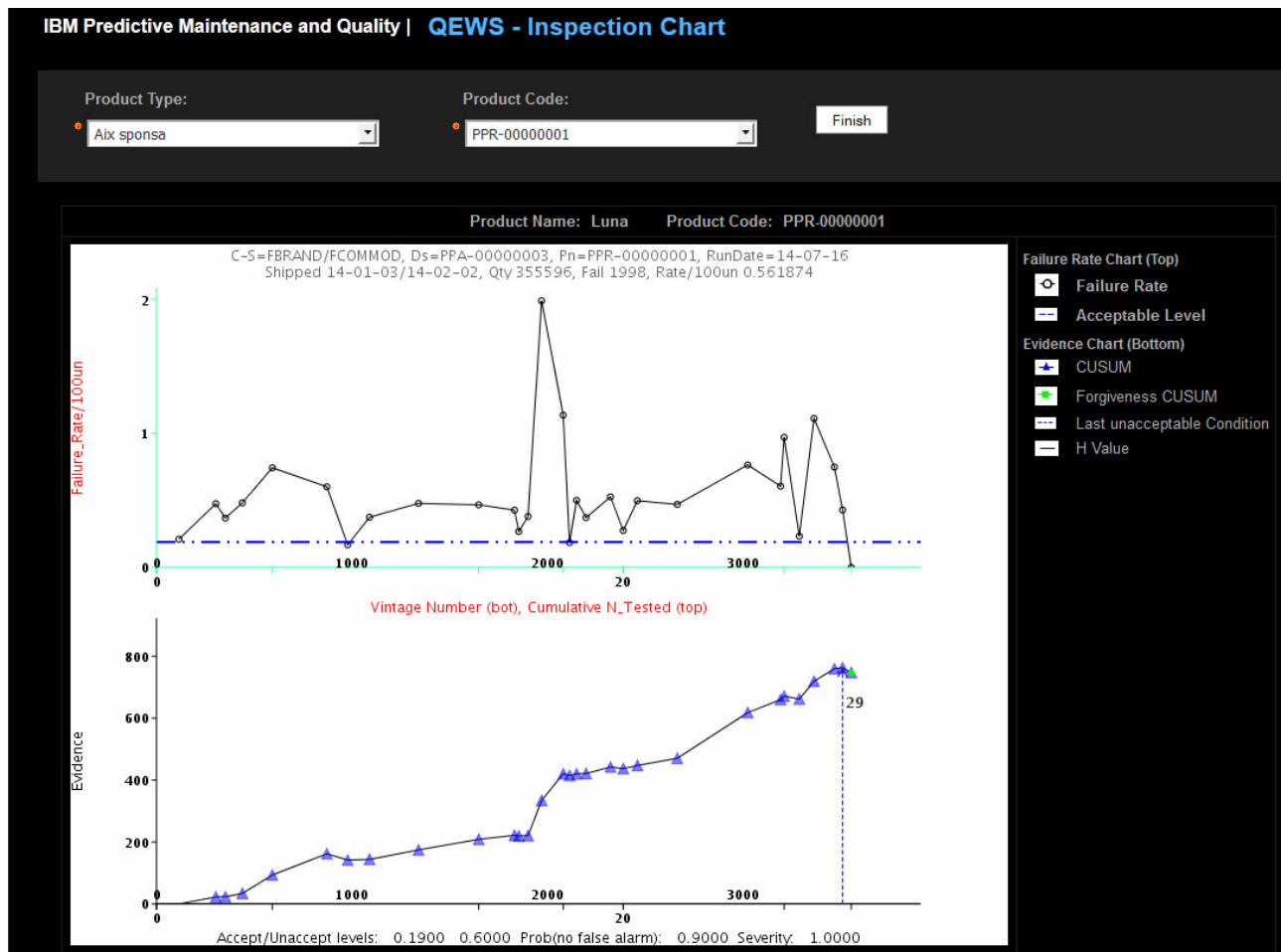


Figure 11-8 QEWS Inspection Chart

The QEWS Inspection Chart includes two charts about failure and evidence:

- Failure Chart
- Evidence Chart

Failure Chart

The Failure Chart appears at the top, and has these details:

- Y Axis: Shows the overall failure rate per 100 tested units.
- X Axis: There is a dual X Axis:
 - Vintage Number (bottom): Indicates the day on which the part was produced during the reporting period.
 - Cumulative N Tested (Top): Indicates the number of parts that were tested for various vintages (dates).
- The blue dotted horizontal line indicates the Acceptable level of failures.

Evidence Chart

The Evidence Chart appears at the bottom, and has these details:

- ▶ Y Axis: Shows the cumulative sum (CUSUM)
- ▶ X Axis: There is a dual X Axis:
 - Vintage Number (bottom): Indicates the day that the part was produced during the reporting period.
 - Cumulative N Tested (top): Indicates the number of parts that were tested for various vintages (dates).
- ▶ The black horizontal line indicates the threshold referred to in QEWS literature as the *H Value*. The Markers (triangles) for CUSUM are shown only when the CUSUM values are above this threshold line.
- ▶ The blue vertical dotted line indicates the vintage on which the last unacceptable condition was recorded for the product during the time period.
- ▶ The Green marker represents the forgiveness factor of a selected product or part.
- ▶ Rundate is the date when data was analyzed by QEWS.

Figure 11-9 gives summary of QEWS-generated results.

C-S = FBRAND/FCOMMOD		Ds = PPA-00000003		Pn = Luna		Run Date = Jul 16, 2014	
Shipped = Jan 3, 2014 / Feb 2, 2014		Qty: 59,266.00		Fail: 333.00		Rate / 100un: 0.56	
Accept / Unaccept Levels: 0.19 / 0.6		Prob (No False Alarm) : 0.9		H Value: 0.74			

OBS	DATES	Cumulative N_Tested	CUSUM	TESTED %	FAILS	RATES	OV	QTY	%FAIL	%TSTC	%FAILC	%FVIN%	Last Unacceptable Condition	Forgiveness CUSUM
1	Jan 3, 2014	114.24	0.00	19.04	4.00	0.210	0	1,904	0.20	3.20	1.20	1.20		
2	Jan 4, 2014	303.54	22.50	31.55	15.00	0.475	0	3,155	0.50	8.50	5.70	4.50		
3	Jan 5, 2014	352.44	23.07	8.15	3.00	0.368	0	815	0.40	9.90	6.60	0.90		
4	Jan 6, 2014	439.44	34.05	14.50	7.00	0.483	0	1,450	0.50	12.40	8.70	2.10		
5	Jan 7, 2014	592.68	93.41	25.54	19.00	0.744	1	2,554	0.70	16.70	14.40	5.70		
6	Jan 8, 2014	871.74	161.91	46.51	28.00	0.602	1	4,651	0.60	24.50	22.80	8.40		
7	Jan 9, 2014	978.42	141.88	17.78	3.00	0.169	0	1,778	0.20	27.50	23.70	0.90		
8	Jan 10, 2014	1,090.32	143.98	18.65	7.00	0.375	0	1,865	0.40	30.70	25.80	2.10		
9	Jan 11, 2014	1,341.30	174.49	41.83	20.00	0.478	0	4,183	0.50	37.70	31.80	6.00		
10	Jan 12, 2014	1,649.28	208.68	51.33	24.00	0.468	0	5,133	0.50	46.40	39.00	7.20		
11	Jan 13, 2014	1,831.86	221.58	30.43	13.00	0.427	0	3,043	0.40	51.50	42.90	3.90		
12	Jan 14, 2014	1,854.24	219.60	3.73	1.00	0.268	0	373	0.30	52.10	43.20	0.30		
13	Jan 15, 2014	1,901.58	220.72	7.89	3.00	0.380	0	789	0.40	53.50	44.10	0.90		
14	Jan 16, 2014	1,970.94	333.99	11.56	23.00	1.990	1	1,156	2.00	55.40	51.10	6.90		
15	Jan 17, 2014	2,081.76	420.48	18.47	21.00	1.137	1	1,847	1.10	58.50	57.40	6.30		
16	Jan 18, 2014	2,114.16	414.93	5.40	1.00	0.185	0	540	0.20	59.50	57.70	0.30		
17	Jan 19, 2014	2,150.04	420.13	5.98	3.00	0.502	0	598	0.50	60.50	58.60	0.90		
18	Jan 20, 2014	2,198.46	420.87	8.07	3.00	0.372	0	807	0.40	61.80	59.50	0.90		
19	Jan 21, 2014	2,323.74	442.20	20.88	11.00	0.527	0	2,088	0.50	65.30	62.80	3.30		
20	Jan 22, 2014	2,389.08	436.90	10.89	3.00	0.275	0	1,089	0.30	67.20	63.70	0.90		

Figure 11-9 Inspection Summary report

11.3 Warranty analytics

Warranty analytics is an implementation of the advanced QEWSL algorithm, which is designed to give early warning signals about accelerated wear or replacement indications for parts, with minimal false alarms.

Warranty analytics covers three primary use cases: Sales, production, and manufacturing. The master data required for processing warranty data is loaded using the master data flows explained earlier in this book.

Table 11-2 provides additional details about the warranty analytics use cases.

Table 11-2 Warranty analytics use cases

Use case	Details
Sales	This use case deals with the sale of products that are governed under warranty terms. In PMQ, products are tracked through resource and production batches. The most important criteria for master data in this use case is the mapping between these batches in the MASTER_RESOURCE_PRODUCTION_BATCH table. This table also maintains the quantity information about the number of parts and products that were used to build the resource.
Production	This use case deals with warranty analysis, with a focus on the production date of the product. The most important criteria for master data in this use case is the production date (produced_date) information that is stored in the MASTER_PRODUCTION_BATCH table, which is used as the vintage date.
Manufacturing	This use case deals with warranty analysis based on the manufacturing date of any resource that contains monitored parts or products. A resource is considered to be tracked for warranty purposes based on the packaging or manufacturing date. The manufacturing date is stored in the MASTER_RESOURCE table. The warranty for the resource starts on the manufacturing date and goes until the defined warranty period ends.

The sequence of warranty analytics processing is as shown in Figure 11-10. More details are provided in the next subsections.

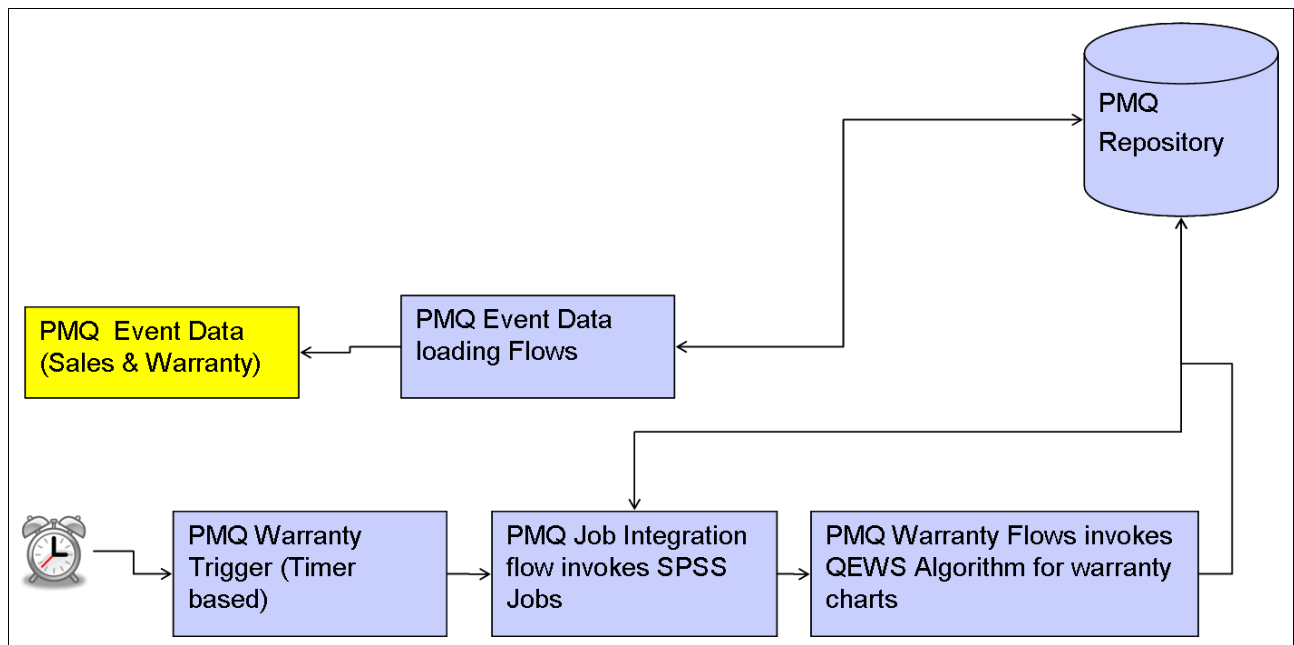


Figure 11-10 Warranty analytics processing sequence

11.3.1 Data preparation for warranty analytics

Warranty data is reported as events. Two types of warranty-related events are reported: Sales events and warranty events.

Sales events

Sales events provide observations such as sale date and warranty period (in months). These events are processed based on the orchestration rules defined in the Sales orchestration definition XML shown in Figure 11-11.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_mapping	
event_orchestration_key_cd	sales
orchestration_cd	pmq.sales
orchestration	
orchestration	
orchestration_cd	pmq.sales
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	

Figure 11-11 Sales orchestration definition XML

A Sales event has only one orchestration step:

- ▶ The EventStoreAdapter persists Sales events into the EVENT and EVENT_OBSERVATION tables.

Warranty events

Warranty events provide observations such as claim date and a so-called warranty flags that declare whether the product or part is covered under a warranty. These events are processed based on the orchestration rules defined in the Warranty orchestration definition XML shown in Figure 11-12.

?? xml	version="1.0" encoding="UTF-8"
ns3:orchestration_definition	
xmlns:Q1	http://www.ibm.com/asf
xmlns:ns3	http://www.ibm.com/asf/orchestration
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/asf/orchestration orchestration_definition.xsd
event_orchestration_mapping	
event_orchestration_key_cd	warranty
orchestration_cd	pmq.warranty
event_orchestration_mapping	
orchestration	
orchestration_cd	pmq.warranty
step	
adapter_class	com.ibm.analytics.foundation.adapter.eventstore.EventStoreAdapter
adapter_configuration	
orchestration	

Figure 11-12 Warranty orchestration definition XML

A Warranty event has only one orchestration step:

- The EventStoreAdapter persists Warranty events into the EVENT and EVENT_OBSERVATION tables.

11.3.2 SPSS Modeler streams

For warranty analytics, data preparation involves two SPSS streams, IBMPMQ_QEWSL_SALES.str for Sales events and IBMPMQ_QEWSL_WARR.str for warranty events. These SPSS streams read data from the EVENT, EVENT_OBSERVATION, MASTER_RESOURCE, MASTER_PRODUCT, and MASTER_PRODUCTION_BATCH tables and prepare that data in a specific format:

```
Produced Date | Product_Id | Service_Month | Parts under Warranty | Parts replaced  
| Tenant_Id
```

This prepared data is stored in a staging table called SERVICE. The records of this table are fed as input to the QEWSL algorithm.

The SPSS Modeler streams involved in warranty analytics each have corresponding IBM SPSS Collaboration and Deployment Services (C&DS) jobs. The IBMPMQ_QEWSL_WARR.str stream is used also for the manufacturing and production use cases (controlled by toggling the job parameter IsMFG_OR_PROD from MFG (for manufacturing) to PROD (for production). The IBMPMQ_QEWSL_SALES.str stream is dedicated to the Sales use case and is started by a separate job, IBMPMQ_QEWSL_SALES_JOB. Both jobs can be triggered through IBM Integration Bus message flows.

11.3.3 SPSS customizable parameters

The SPSS Modeler streams used in warranty analytics contain some common parameters that are used while running the SPSS models. These options can be altered from within SPSS C&DS or from Integration Bus (preferred). The next subsections describe the parameters.

IsRunDateEqServerDate

This parameter determines whether the SPSS server system date (value = 1) or a custom-run date (value = 0) is used for computing the run date. The default value is 0, in which case the custom run date that is supplied by the Integration Bus message flow (corresponding to the Integration server system date during default runs) is used.

RunDateInFormatYYYYMMDDHyphenSeparated

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

ServiceTblQtyMultiplier

Sometimes, due to the lack of complete data, or for preliminary analysis, or for performance reasons, you might need to run the QEWSL algorithm on a subset of the data. QEWSL is a weighted algorithm, so by default it does not produce the same graphs, outputs, or alerts for a sample as it would produce for the complete data. If the sample adequately represents the larger, complete set, this parameter (ServiceTblQtyMultiplier) helps to correct the scale of the weighted results to give a representative output. The parameter is set with a multiplier value that is either equal to the size of the complete data set or the size of the sample data available.

11.3.4 SPSS model invocation

The PMQ solution uses Integration Bus message flows to start the SPSS jobs explained previously. The SPSS jobs can be triggered in two ways:

- Placing a text file that contains the run date in the /batchdata\in directory of the Integration Bus node (WarrantyFileIntegration message flow).
- Using a timer mode that sets the current date as the run date (WarrantyTimer message flow).

WarrantyTimer message flow

This flow that is shown in Figure 11-13 is scheduled to run once each day. This flow prepares a WebSphere MQ message with the current date as the run date parameter and then triggers the WarrantyJobIntegration flow to start the SPSS job.

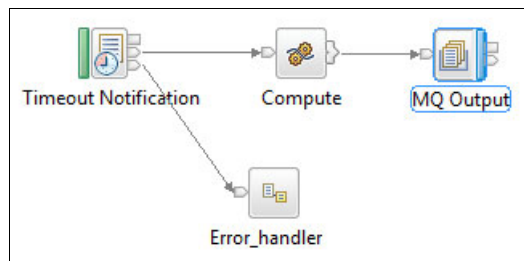


Figure 11-13 WarrantyTimer message flow

WarrantyDataPreparation message flow

The flow that is shown in Figure 11-14 triggers the warranty-related SPSS job by placing a file named qewsrundate (containing the rundate parameter) in the /batchdata\in directory.

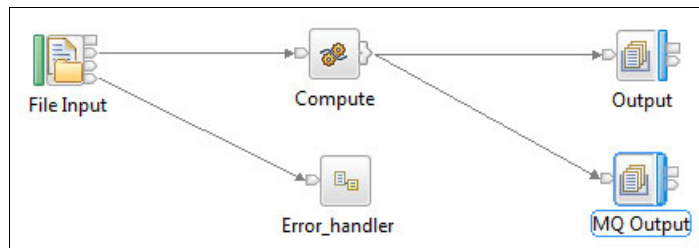


Figure 11-14 WarrantyDataPreparation message flow

WarrantyJobIntegration message flow

The flow shown in Figure 11-15 starts the SPSS warranty-related streams. There are two streams for the Sales, Production, and Manufacturing use cases. Switching to a specific stream is governed by a user-defined property called `UseCaseSelection` within this flow.

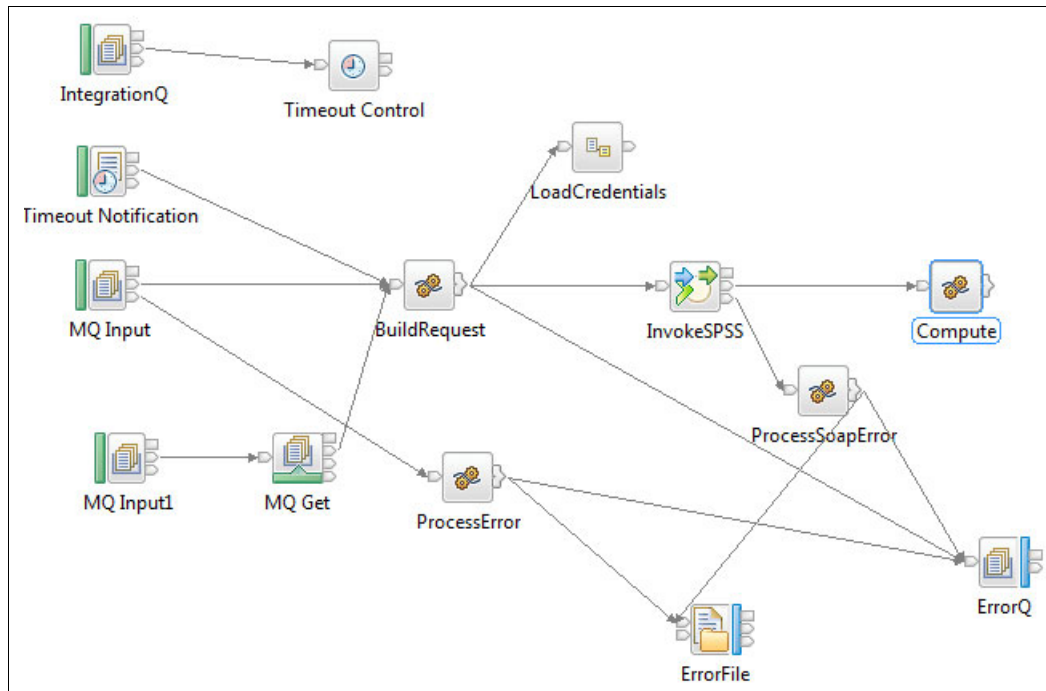


Figure 11-15 *WarrantyJobIntegration* message flow

WarrantyFileIntegration message flow

After the SPSS Modeler stream loads data into the `SERVICE` table, SPSS creates a status file and a `rundate` value, and places them in the `/integrationin` directory of the Integration Bus node. The `WarrantyFileIntegration` message flow shown in Figure 11-16 reads this status file. If the status is `Failure`, further processing is canceled. If the status is `Success`, the `ProcessWarranty` flow is triggered through placement of a message in its input queue.

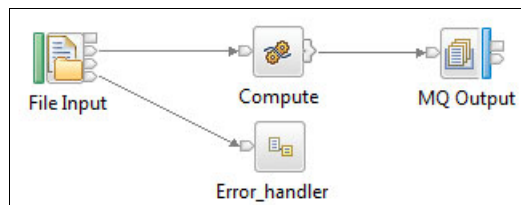


Figure 11-16 *WarrantyFileIntegration* message flow

ProcessWarranty message flow

The ProcessWarranty flow shown in Figure 11-17 reads the rundate from the status message just described. It then queries the SERVICE table and prepares object structures that include the parts under warranty (WPARTS), the parts replaced under warranty (WREPL), the date the produce was produced, the product ID, and various product parameters. These object structures are used to start the QEWSL algorithm.

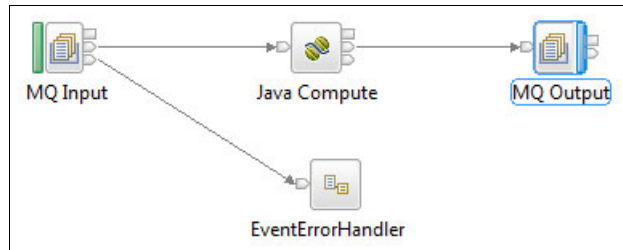


Figure 11-17 ProcessWarranty message flow

The output of the QEWSL algorithm is stored in the Integration Bus node file system and then copied through network file sharing to the Cognos Business Intelligence node. The Integration Bus node file system location is configured in the `loc.properties` properties file in the Integration Bus shared classes directories. Within the `loc.properties` file are two variables, `Location` and `Location1`. `Location1` is where the QEWSL algorithm output is stored. Its value forms the base path where a folder with the rundate value (in the format `yyyy_MM_dd`) is created. Extra folders are created within this folder for each product ID (a combination of `Product_cd` and `Product_type_cd`) with the output files generated by the QEWSL algorithm.

11.3.5 Warranty Analytics report

The Warranty Analytics report includes QEWSL warranty charts. These charts are generated by the QEWSL algorithm and are shown in Figure 11-18. The charts are used for monitoring data and visualizing information about the lifetimes of systems and components.



Figure 11-18 QEWSL warranty charts

There are two QEWSL warranty charts: Replacement Rate and Evidence Chart.

Replacement Rate Chart

The Replacement Rate Chart is at the top. Here are some details about it:

- ▶ It has a dual X Axis that indicates Vintage Number and Cumulative_Number_Tested.
 - Vintage Number indicates the day that the part was produced during the reporting period.
 - Cumulative_Number_Tested indicates the number of parts that were tested.
- ▶ The Y axis indicates the replacement rate of the product (per machine month).
- ▶ A data point on the chart indicates the replacement rate for a vintage number.
- ▶ Acceptable Level is indicated by a horizontal threshold line on the chart that declares the acceptable replacement rate.
- ▶ The chart also depicts a product's Monthly Wearout indexes (the second curved line shown in pink).

Evidence Chart

The Evidence Chart is at the bottom. Here are some details about it:

- ▶ This chart monitors the lifetime of a product.
- ▶ It has a dual X Axis that indicates Vintage Number and Cumulative_Number_Tested.
 - Vintage Number indicates the day that the part was produced during the reporting period.
- ▶ Cumulative_Number_Tested indicates the number of parts that were tested.
- ▶ The Y axis indicates the cumulative sum (CUSUM) value for the product.
- ▶ Threshold H is a horizontal line that indicates the replacement rate threshold value.
- ▶ The CUSUM values that are higher than Threshold H are displayed as triangles on the chart. The triangles indicate that the process is not stable.
- ▶ The vertical dotted line indicates the vintage (date) on which the last unacceptable condition was recorded for the product during the time period.

Figure 11-19 provides the summary information of the QEWS-generated values.

C-S = ADAPTERS/CONTROLLER		Ds = PPX-00000006		Pn = Aquamarine		RUN DATE = May 28, 2014		Severity: 1	
Shipped = Jan 1, 2011 / Mar 20, 2012		Qty: 450.00		Fail: 255.00		Rate/100un: 56.66666667		Sevwear: 1	
Accept / Unaccept Levels: 0.003 / 0.05722		Threshold H: 0.0000		Threshold W: 19.9384					

OBS	DATES	EVIDENCE	CUSUM Wearout	TESTED	FAILS	RATES	OV	QTY	%FAIL	%TSTC	%FAILC	%FVIN%	Last Unacceptable Condition
1	Jan 1, 2011	22.84	31.37	1,848	24	0.013	0	60	40.00	13.30	9.40	9.40	
2	Feb 1, 2011	324.86	1,496.06	1,092	30	0.027	0	60	50.00	26.70	21.20	11.80	
3	Feb 28, 2011	351.87	1,836.92	1,260	54	0.043	0	60	90.00	40.00	42.40	21.20	
4	Mar 30, 2011	326.22	2,533.18	396	54	0.136	1	60	90.00	53.30	63.50	21.20	
5	May 1, 2011	192.52	2,522.89	1,662	18	0.011	0	60	30.00	66.70	70.60	7.10	
6	Dec 29, 2011	287.52	3,212.00	1,350	18	0.013	0	60	30.00	80.00	77.60	7.10	
7	Jan 31, 2012	290.83	3,219.19	1,044	48	0.046	0	60	80.00	93.30	96.50	18.80	
8	Feb 3, 2012	23.01	19.69	121	8	0.066	1	10	80.00	95.60	99.60	3.10	
9	Mar 16, 2012	20.14	0.00	210	1	0.005	0	10	10.00	97.80	100.00	0.40	
10	Mar 20, 2012	16.10	0.00	230	0	0.000	0	10	0.00	100.00	100.00	0.00	

Figure 11-19 Warranty Summary Report



Integrating IBM Maximo Asset Management

This chapter describes integrating the IBM Predictive Maintenance and Quality solution with IBM Maximo Asset Management. Maximo, an enterprise asset management system, is not included with PMQ, but can be integrated with it. Several special adapters are included with the solution for this purpose.

Integration is ideal for companies that have already installed Maximo and want to use it to generate the maintenance work orders that result from the PMQ solution's predictive models, or if they want to feed earlier work orders into the solution to derive new predictive insights to optimize future maintenance schedules.

This chapter covers only the highlights of Maximo integration. More information is available in the Predictive Maintenance and Quality solution documentation located here:

<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

The chapter contains these sections:

- ▶ Synchronization of solution data with Maximo
- ▶ Consuming recommended actions in Maximo

12.1 Synchronization of solution data with Maximo

An automated approach using message flows has been designed to synchronize the data between the solution and Maximo.

The mapping between the Predictive Maintenance and Quality solution and Maximo works under these principles:

- ▶ The CLASSSTRUCTURE object structure of Maximo is mapped to the master_group_dim entity of the solution. The records in the group_dim table provide classifications for resources. There can be up to five classifications for each resource and each classification can vary.
- ▶ The SERVICEADDRESS object structure of Maximo is mapped to the master_location entity of the solution. The location table contains the location of a resource or event, such as a room in a factory or a mining site. In Maximo, this information is stored as a LOCATIONS object and in the associated SERVICEADDRESS object.
- ▶ The ASSET object structure of Maximo is mapped to the master_resource entity of the solution. Asset information imported from Maximo includes the asset type, classification, and location. A resource defines resources of types asset or agent. An asset is a piece of equipment, whereas the agent is the operator of the equipment. Some asset resources can form a hierarchy (for example, a truck is a parent of a tire).

When there is a change in the master data in Maximo, it is exported with the help of a Publish channel that is configured in Maximo.

The Predictive Maintenance and Quality solution includes the message flow shown in Figure 12-1 to process the ASSET, SERVICEADDRESS, and CLASSSTRUCTURE object structures of Maximo. The solution also provides mapping routines to transform the Maximo object structure data to the format required by the solution. Any errors that are encountered during mapping or data-loading activities generate an error file in the Error directory. The file includes the cause of the error, if further investigation is needed.

Data loading is supported in both batch and real-time modes. Batch mode is used mostly to export existing (historical) data, whereas real-time mode is used for new or updated data. In batch mode, master data must be exported from Maximo in XML format and must be placed into the \maximointegration folder. In real-time mode, master data gets updated using a SOAP request that is sent to the solution, where the changes are replicated.

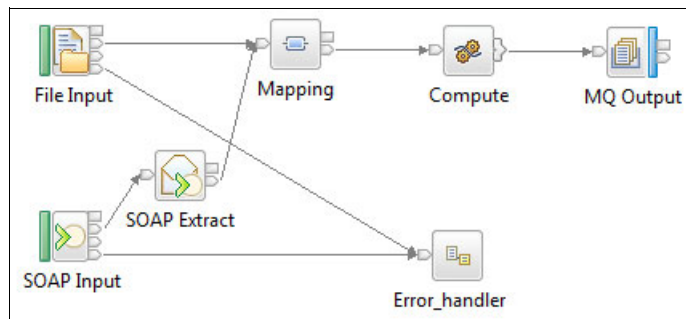


Figure 12-1 Master data message flow in Maximo

Maximo work orders are mapped as events in the PMQ solution. The WORKORDER object structure in Maximo is mapped to the solution's event structure. The solution only maps specific work orders such as BREAKDOWN, Actual Maintenance, and Scheduled Maintenance. If any additional mapping is needed, the message flows must be modified.

Errors that are encountered during mapping or data loading activities are written to the /error directory.

After work orders are mapped as events, the solution processes them through standard event processing flows (see Figure 12-2 and Figure 12-3). Work order data can be synchronized in the solution using both batch and real-time modes. The process is similar to that described for master data earlier in this chapter.

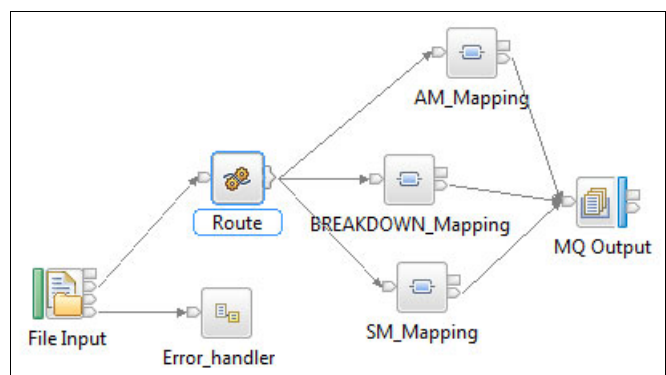


Figure 12-2 Work order mapping flow in batch mode

Figure 12-3 shows the work order mapping flow when using real-time mode.

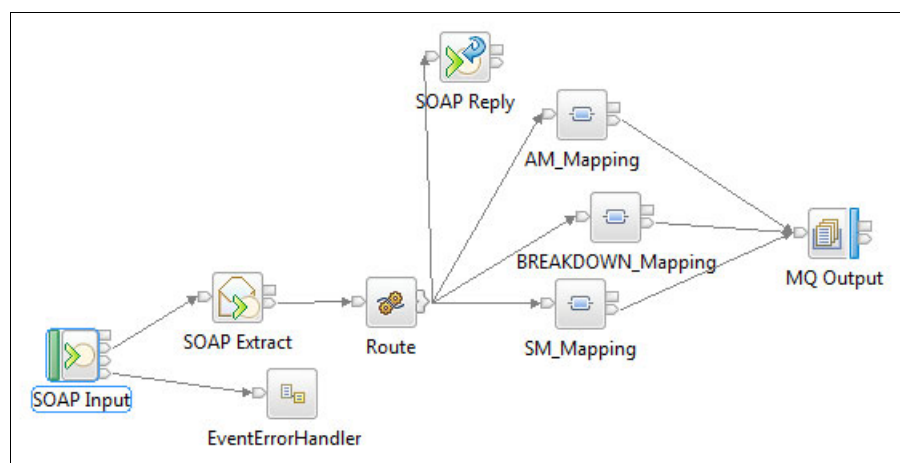


Figure 12-3 Work order mapping flow in real-time mode

12.2 Consuming recommended actions in Maximo

The solution generates maintenance recommendations whenever an alarming situation develops with a particular asset, such as when performance of the asset is obviously deteriorating. These recommendations are based on the solution's business rules and predictive models, and can act as a trigger for creating a work order in Maximo.

To create a work order in Maximo, the solution uses the EXTSYS1_MXWOInterface service, which is exposed by Maximo and started by the solution. In addition, a Maximo web service must be configured to correspond with the service defined in the MaximoWorkOrder.wsd1 file in the PMQMaximoIntegration application.

The solution uses IBM Integration Bus flows to start the Maximo work order web service, as shown in Figure 12-4.

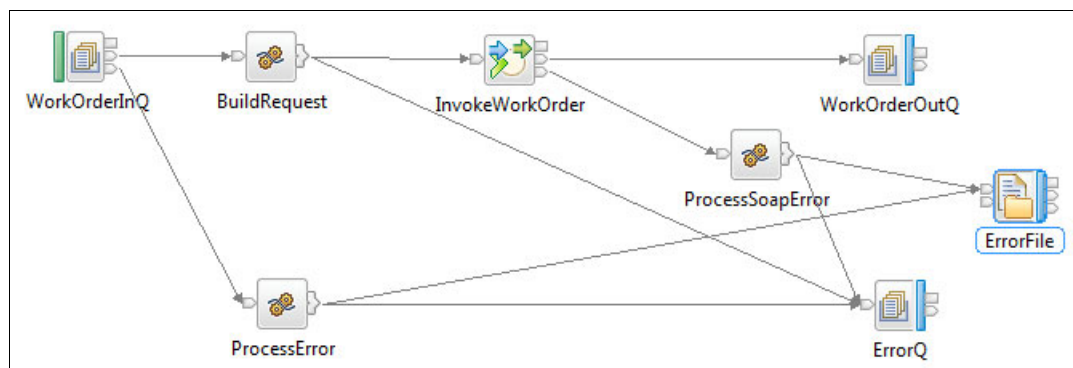


Figure 12-4 Work order creation message flow

Message flows in the Predictive Maintenance and Quality solution can be configured to enable or disable work order creation and updating in Maximo. This is done by updating orchestration definition XML. The message flow shown in Figure 12-4 is used both for creating a work order and updating an existing order. If an error is encountered while creating or updating the work order, an appropriate error message is written to the error queue.

Example 12-1 shows an XML request structure for a work order.

Example 12-1 XML request structure for Maximo work order

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:max="http://www.ibm.com/maximo">
<soapenv:Header/>
<soapenv:Body>
<max:SyncMXWO>
<max:MXWOSet>
<!--Zero or more repetitions:-->
<max:WORKORDER action="AddChange">
<!--Optional:-->
<max:SITEID>BEDFORD</max:SITEID>
<!--Optional:-->
<max:ASSETNUM>SP_MACHINE1</max:ASSETNUM>
<!--Optional:-->
<max:DESCRIPTION>Urgent Inspection</max:DESCRIPTION>
</max:WORKORDER>
</max:MXWOSet>
</max:SyncMXWO>
</soapenv:Body> </soapenv:Envelope>
  
```

For more information about creating an enterprise service, see the IBM Maximo Asset Management Knowledge Center at this location:

<http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp>



Additional reports

This chapter provides an overview of the reports and dashboards available in the IBM Predictive Maintenance and Quality solution. Several reports associated with specific business use cases have already been covered in previous chapters.

The solution is integrated with the IBM Cognos product suite, so you can develop and manage reports and dashboards using Cognos Business Intelligence, Cognos Reporting Studio, and Cognos Framework Manager.

This chapter includes the following sections:

- ▶ Reporting concepts
- ▶ Predictive Maintenance and Quality reports

13.1 Reporting concepts

Reporting is vital to any predictive maintenance solution, and the best solutions provide not only reports, but intuitive, real-time dashboards that take traditional information display to a new level. Conventional means are not sufficient to derive insight from large volumes of data (up to billions of rows) that come from different sources and have multiple attributes and dimensions. Businesses today prefer visually rich, interactive, and dynamic reports that contain both aggregated and detailed data to support the needs of different users.

Reports and dashboards typically provide these standard features:

- ▶ Dynamic data content with limited data latency
- ▶ Advanced graphs, charts, and gauges
- ▶ Visual querying and exploration of data
- ▶ Visualization of static data from multiple dimensions
- ▶ Personalization tools to customize data views

The Predictive Maintenance and Quality solution goes beyond these standard features and provides a set of advanced, ready-made dashboards and reports that include:

- ▶ Site Overview Dashboard
- ▶ Equipment reports
- ▶ Product Quality Dashboard
- ▶ Audit Report
- ▶ Material Usage by Production batch
- ▶ Advanced KPI Trend Chart

Users can customize and extend these pre-configured dashboards and reports, or create entirely new ones based on their specific needs. The metadata that defines each report and dashboard can be modified by using IBM Cognos Framework Manager.

Details about how the solution uses Cognos Framework Manager are provided in the *Predictive Maintenance and Quality Solution Guide* available with other solution documentation at this location:

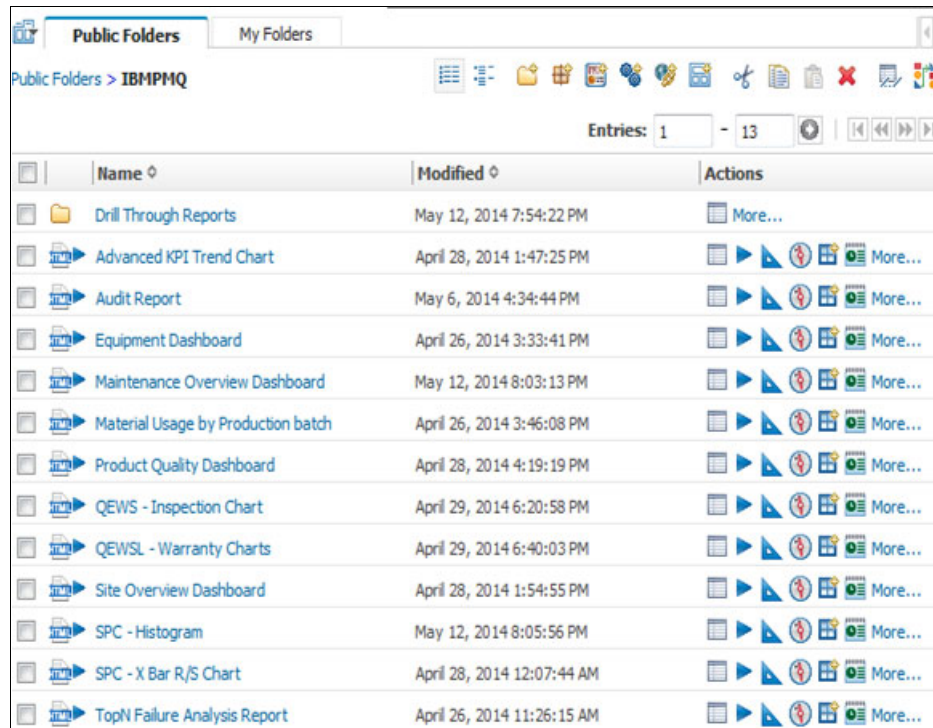
<http://www-01.ibm.com/support/docview.wss?uid=swg27041633>

13.2 Predictive Maintenance and Quality reports

The Predictive Maintenance and Quality solution integrates with IBM Cognos products to enable various real-time views into device status, performance, and maintenance needs. This section describes the most useful of these reports, a term that is used here to describe both traditional reports, which are mostly text and numbers, and dashboards, which include more visual elements.

13.2.1 Accessing reports

IBM Cognos Report Studio is used to access and visualize your preventive maintenance data. After you access the Cognos server and select the appropriate project folder, a list of available reports is displayed in the window (Figure 13-1). To view a report, click the hyperlink for it.



The screenshot shows the IBM Cognos Report Studio interface. At the top, there are tabs for 'Public Folders' and 'My Folders'. Below the tabs, the breadcrumb path is 'Public Folders > IBMMPMQ'. A toolbar with various icons is visible. Below the toolbar, there is a table with columns 'Name', 'Modified', and 'Actions'. The table lists 13 items, including folders and reports. Each item has a 'More...' link in the 'Actions' column.

Name	Modified	Actions
Drill Through Reports	May 12, 2014 7:54:22 PM	More...
Advanced KPI Trend Chart	April 28, 2014 1:47:25 PM	More...
Audit Report	May 6, 2014 4:34:44 PM	More...
Equipment Dashboard	April 26, 2014 3:33:41 PM	More...
Maintenance Overview Dashboard	May 12, 2014 8:03:13 PM	More...
Material Usage by Production batch	April 26, 2014 3:46:08 PM	More...
Product Quality Dashboard	April 28, 2014 4:19:19 PM	More...
QEWS - Inspection Chart	April 29, 2014 6:20:58 PM	More...
QEWSL - Warranty Charts	April 29, 2014 6:40:03 PM	More...
Site Overview Dashboard	April 28, 2014 1:54:55 PM	More...
SPC - Histogram	May 12, 2014 8:05:56 PM	More...
SPC - X Bar R/S Chart	April 28, 2014 12:07:44 AM	More...
TopN Failure Analysis Report	April 26, 2014 11:26:15 AM	More...

Figure 13-1 List of available reports and dashboards in IBM Cognos Report Studio

The list of reports is presented alphabetically by default, but it is useful to understand the report hierarchy from a functional perspective. This organization is depicted in Figure 13-2.

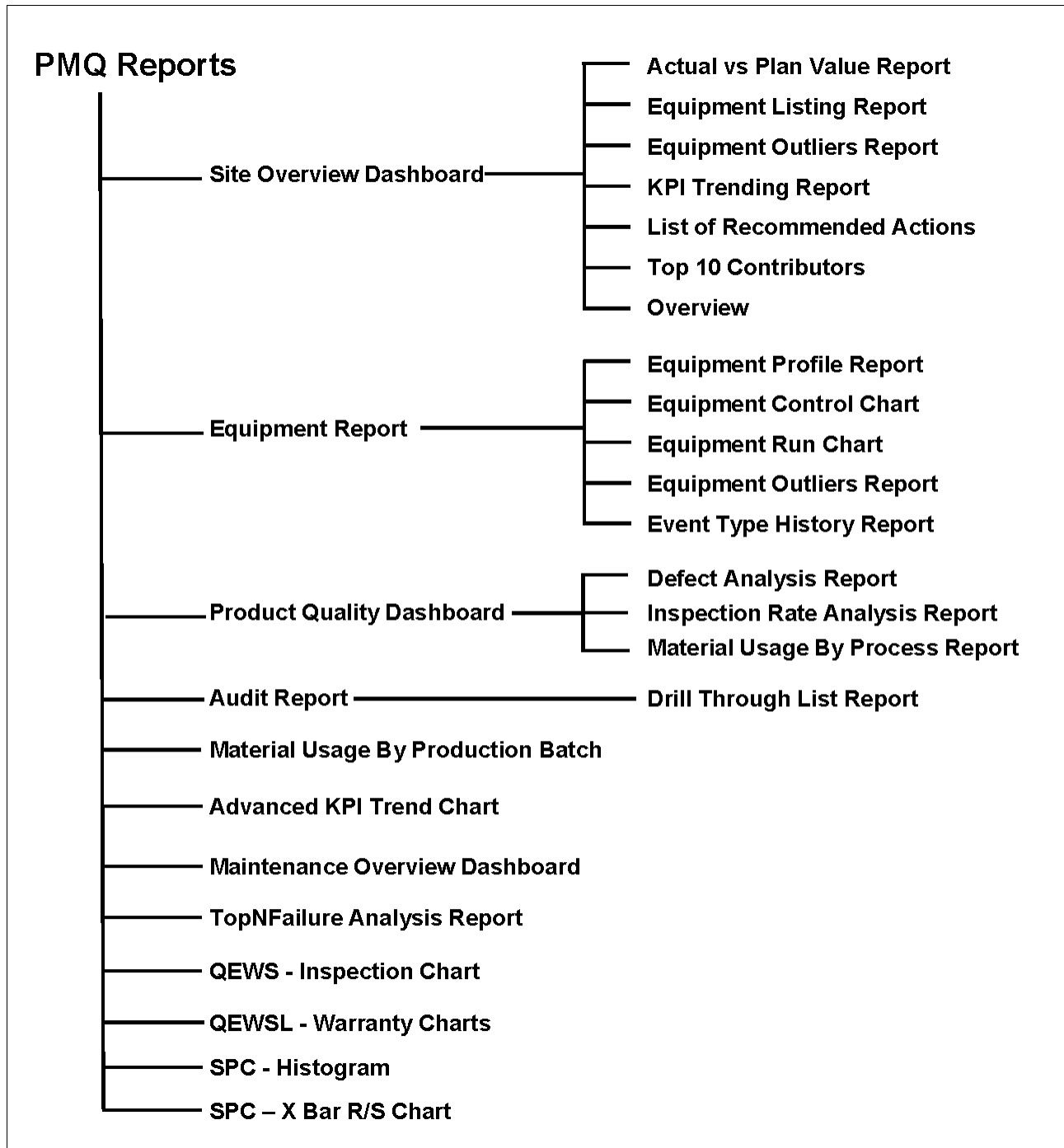


Figure 13-2 Hierarchy of available predictive maintenance reports

13.2.2 Site Overview Dashboard

The Site Overview Dashboard (Figure 13-3) provides a high-level summary of all monitored assets at all monitored sites. The dashboard shows the key performance indicators (KPIs) considered to have the greatest impact on asset maintenance. These include the KPIs for Health Score, Resource Count, MTTR (Mean Time To Repair), and MTBF (Mean Time Between Failures).

The dashboard can also show charts that provide a high-level summary of the health of all of your assets. These charts are displayed by selecting **Overview** from the **Show** menu. Available charts include these:

- Overview: Health Score Trend (bar chart)
- Overview: Health Score Contributor (pie chart)
- Overview: Incident and Recommendation Summary Report (bar chart)

Filters that can be applied to this dashboard are From Date, To Date, Location, and Resource Sub Type.

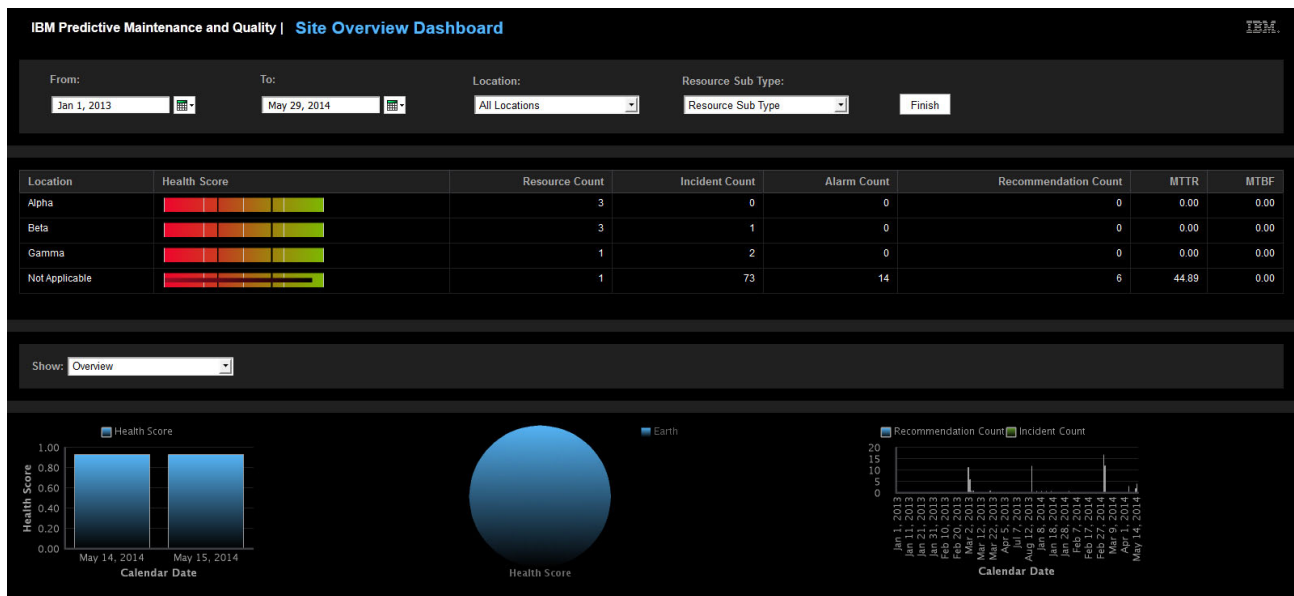


Figure 13-3 Site Overview Dashboard showing all monitored assets

Recommendation Event List

The Recommendation Event List report summarizes all recommended actions and incident details for a piece of monitored equipment. It is accessed by clicking the **Recommendation** or **Incident Analysis** bar chart in the Site Overview Dashboard. The report in Figure 13-4 shows that an urgent inspection was recommended for Transformer 1, probably because its Health Score (0.93) dropped below a designated threshold.

IBM Predictive Maintenance and Quality Recommendation Event List				
Equipment Name	Resource Code	Health Score	Date	Recommended Action
Earth	AAAX2-ZZZZT-TV	0.93	May 14, 2014 11:04:40 PM	Urgent Inspection
	AAAX2-ZZZZT-TV	0.93	May 14, 2014 11:04:40 PM	Urgent Inspection

Figure 13-4 Recommendation Event List for a single monitored asset

The Incident Event List report in Figure 13-5 counts the number of failures that are recorded by resources.

IBM Predictive Maintenance and Quality Incident Event List				
Equipment Name	Resource Code	Measurement	Date	Failure Code
Earth	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 12:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 4:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 8:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 12:00:00 PM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 4:00:00 PM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 8:00:00 PM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 12:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 4:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 8:00:00 AM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 12:00:00 PM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 4:00:00 PM	X101
	AAAX2-ZZZZT-TV	12.00	Mar 3, 2014 8:00:00 PM	X101

Figure 13-5 Incident Event List for a single monitored asset

The filters that can be applied to this report are Date, Location, and Resource Sub Type.

Top 10 Contributors

The Top 10 Contributors report (Figure 13-6) appears in the bottom portion of the site Overview Dashboard, and identifies the top 10 pieces of equipment, locations, and operators responsible for reported failures.

The filters that can be applied to this report are Date, Location, Resource Sub Type, and Measurement Type.

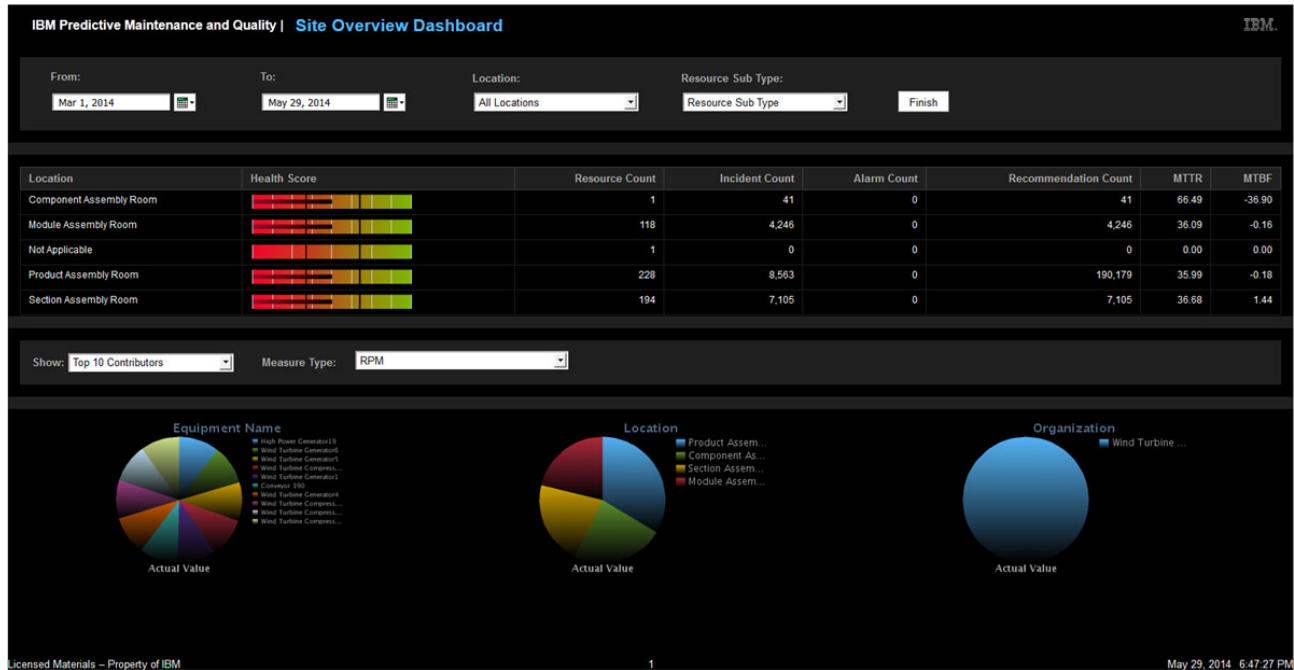


Figure 13-6 Top 10 Contributors report

Outliers

The Outliers report (Figure 13-7) lists equipment or assets that are performing outside of allowable limits. The last recorded value for a profile variable for a particular resource is compared against predefined allowable limits for lifetime to date (LTD) average value and LTD standard deviation.

The filters applied to this report are Date, Location, Resource Sub Type, and Sigma level.

Show:

Equipment Outliers

Sigma:

1

Equipment Type	Resource Code	Profile Variable Name	Avg	Lower Control Limit	Upper Control Limit	Last Value	Profile Units
Aura	AAV5-ZZZZT-TT	Component Load Test	51.14	20.56	81.71	0.05	
	AAV5-ZZZZT-TT	Quantity Produced	46.81	13.01	80.61	4.72	
	AAV5-ZZZZT-TT	Section Test	23.00	7.76	38.24	0.20	
Earth	AAV2-ZZZZT-TV	DEV Measurement of Type	1.71	-2.82	6.25	12.00	
	AAV2-ZZZZT-TV	FAIL Measurement of Type	1.24	-3.04	5.52	43.16	
	AAV2-ZZZZT-TV	Forecasted Days to Next Maintenance	10.43	9.29	11.56	13.00	
	AAV2-ZZZZT-TV	Maintenance Health Score	0.32	0.31	0.34	0.36	
	AAV2-ZZZZT-TV	Operating Hours	51.42	12.50	90.34	104.48	hrs
	AAV2-ZZZZT-TV	RPM	93.74	58.11	129.37	11.14	
	AAV2-ZZZZT-TV	Scheduled Days to Next Maintenance	0.14	-0.24	0.52	1.00	
Hydra	AAV6-ZZZZT-TR	Assembly Time	51.54	20.06	83.02	122.49	hrs
	AAV6-ZZZZT-TR	Component Life Test	54.00	25.28	82.73	4.47	
	AAV6-ZZZZT-TR	Quantity Produced	56.47	25.40	87.54	22.66	
Volcano	AAV4-ZZZZT-TY	Quantity Produced	59.55	30.05	89.05	200.79	
	AAV4-ZZZZT-TY	Section Load Test	48.86	18.78	78.93	0.03	

Figure 13-7 Outliers report

Actual versus Plan Value

The Actual versus Plan Value report provides information about when monitoring activities occur and what measurements are calculated for different resource profiles. It is accessed by selecting **Actual versus Plan** from the **Show** menu.

The report (Figure 13-8) monitors how closely the metrics for a particular asset are being tracked. The last recorded actual value of the resource is compared against the planned value, and any variances are highlighted.

The filters that can be applied to this report are Date, Location, Resource Sub Type, and Profile Variable.

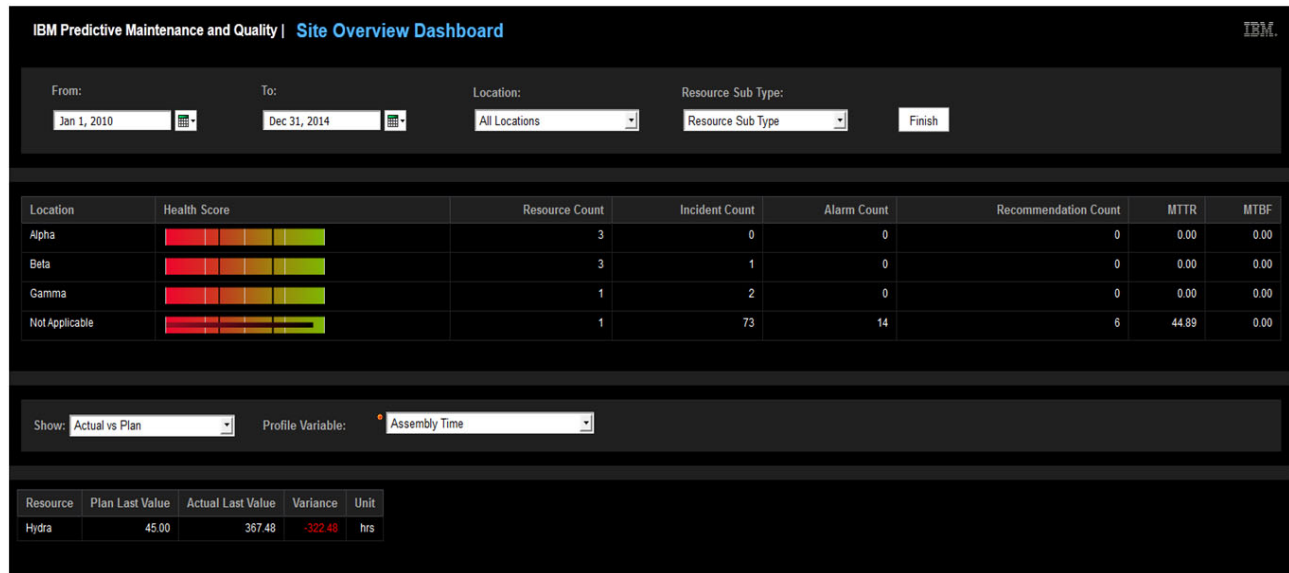


Figure 13-8 Actual versus Plan Value Report

Equipment Listing

The Equipment Listing report (see Figure 13-9) provides a view of all of the monitored equipment, and the health scores and relevant KPIs for that equipment. It is accessed by selecting **Equipment Listing** from the **Show** menu.

The health score for the entire site is derived from the combined health scores of each individual piece of equipment. The scores are presented at the equipment level, such as for a transformer, but the report can be customized to show scores for each separately monitored component within a larger piece of equipment.

The filters that can be applied to this report are Date, Location, and Resource Sub Type.

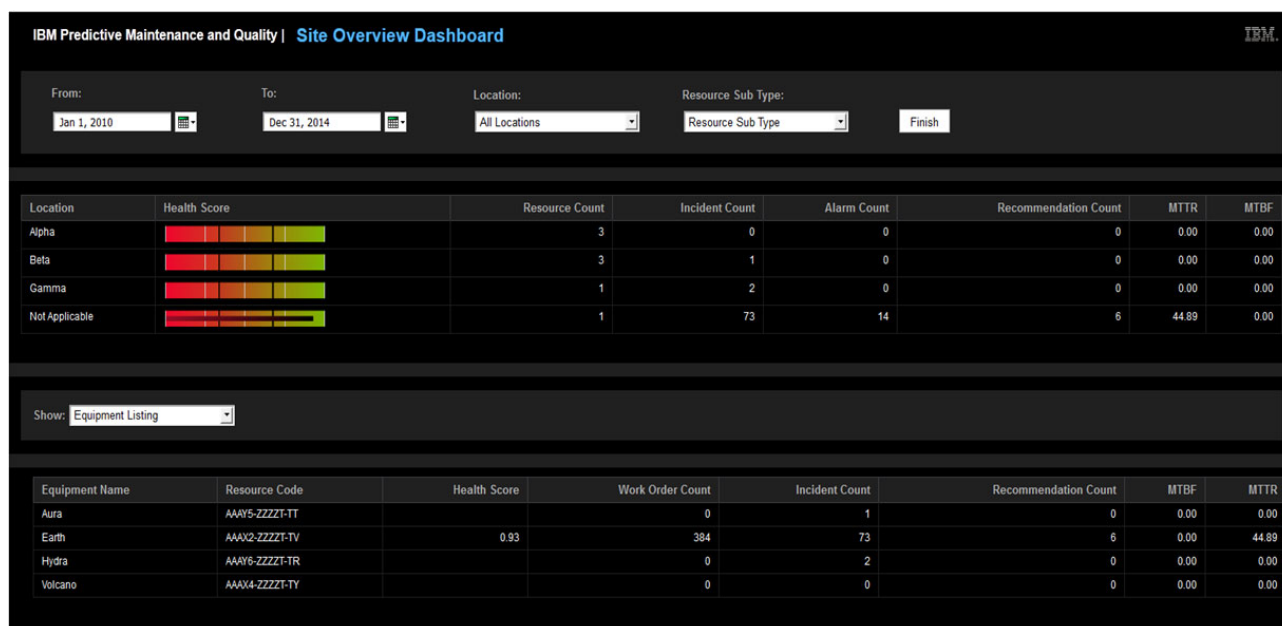


Figure 13-9 Equipment Listing Report showing health scores and KPIs for individual transformers at various sites

KPI Trending

The KPI Trending report allows you to select multiple KPIs to be plotted side-by-side in a line chart. It is accessed by selecting **KPI Trending** from the **Show** menu. The report helps you to find correlations between the selected KPIs and identify any lagging behavior.

For example, if there is a spike in one KPI, how long does it take to affect the other KPIs? Figure 13-10 shows a plot of different condition-monitoring KPIs for an asset taken at different times. By studying the KPI trend for abnormal spikes in value, you can take remedial action before larger problems develop.

The filters that can be applied to this report are Date, Location, Resource Sub Type, Resource, and Measure.

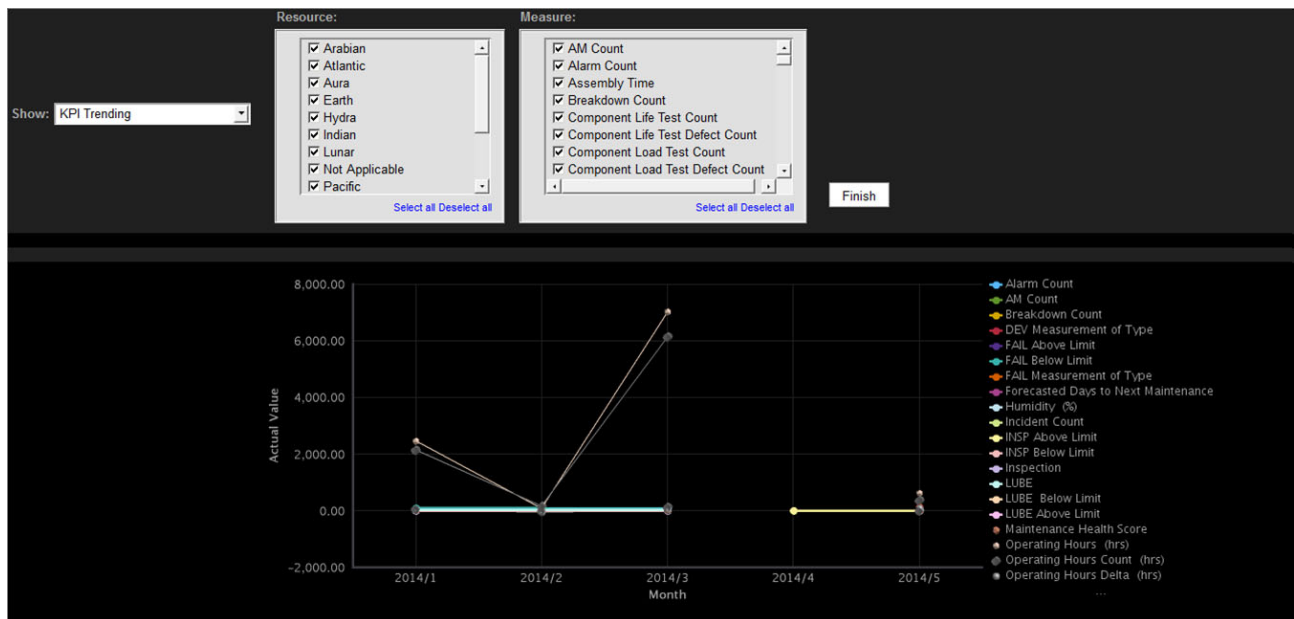


Figure 13-10 KPI Trending Report used to assess asset health

List of Recommended Actions

The List of Recommended Actions report (Figure 13-11) summarizes all recommended actions for each monitored piece of equipment. The report bases its recommendations on the Health Score generated by the solution's predictive models.

The filters that can be applied to this report are Date, Location, Resource Sub Type, and Resource.

Show: List Of Recommended Actions ▾				
Equipment Name	Resource Code	Health Score	Recommended Actions	Date
Earth	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 14, 2014 11:04:40 PM
	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 14, 2014 11:04:40 PM
	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 15, 2014 5:48:17 PM
	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 15, 2014 5:48:17 PM
	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 15, 2014 8:42:40 PM
	AAAX2-ZZZZT-TV	0.93	Urgent Inspection	May 15, 2014 8:42:40 PM
Not Applicable	-NA-	0.93	Urgent Inspection	May 14, 2014 11:04:40 PM
	-NA-	0.93	Urgent Inspection	May 14, 2014 11:04:40 PM
	-NA-	0.93	Urgent Inspection	May 15, 2014 5:48:17 PM
	-NA-	0.93	Urgent Inspection	May 15, 2014 5:48:17 PM
	-NA-	0.93	Urgent Inspection	May 15, 2014 8:42:40 PM
	-NA-	0.93	Urgent Inspection	May 15, 2014 8:42:40 PM

Figure 13-11 List of recommended actions for assets

13.2.3 Equipment Dashboard

Equipment reports include several subordinate reports and charts.

Equipment Profile

The Equipment Profile report shows everything that is known about a piece of monitored equipment, including how it is performing today and how it has performed in the past. The report is accessed by selecting **Equipment Profile** from the **Show** menu.

The report depicted in Figure 13-12 shows the values (past and present) for a particular monitored item.

The filters that can be applied to this report are Resource Sub Type, Resource Name, Resource Serial Number, Location, and Event Code.

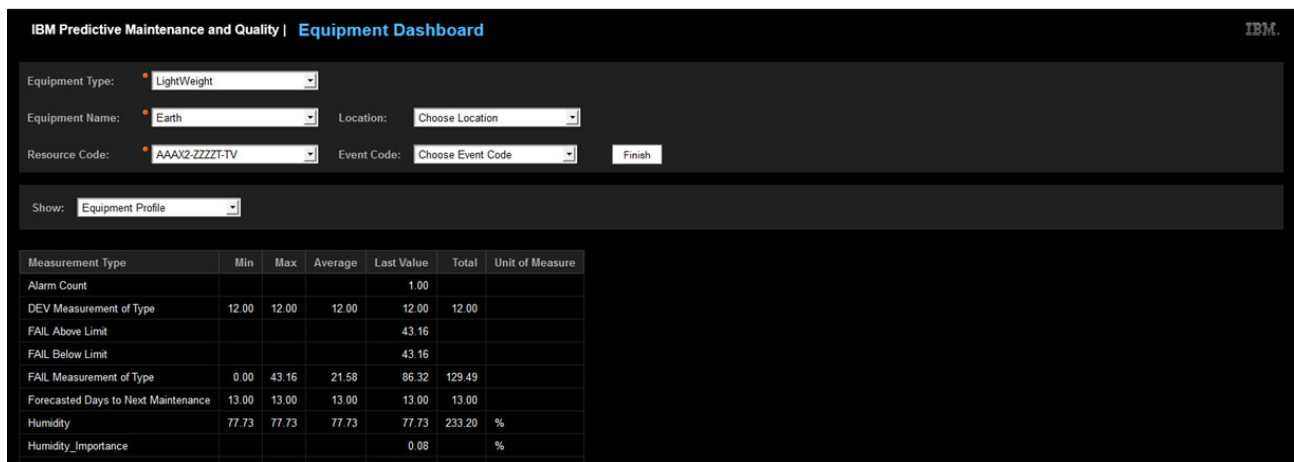


Figure 13-12 Equipment Profile report for a particular asset

Equipment Control Chart

The Equipment Control Chart shows the upper, lower, and the average limit for a selected measurement type relating to a piece of monitored equipment. These limits are maintained in the system in the profile for future reference. The chart is accessed by selecting **Equipment Control Chart** from the **Show** menu.

The Equipment Control Chart depicted in Figure 13-13 shows the likely upper and lower control limits for humidity and variation of humidity at different times of a day.

The limits are calculated in this way:

- ▶ Average (life-to-date) limit: The average limit calculated over the life of the equipment
- ▶ Upper limit: The Average limit + the Sigma Level × the Standard Deviation
- ▶ Lower limit: The Average limit - the Sigma Level × the Standard Deviation

The filters that can be applied to this report are Resource Sub Type, Resource Name, Resource Code, Location, Event Code, Calendar Date, Start Time, End Time, Measurement Type, Profile Variable, and Sigma Level (the spread or variation in the values).

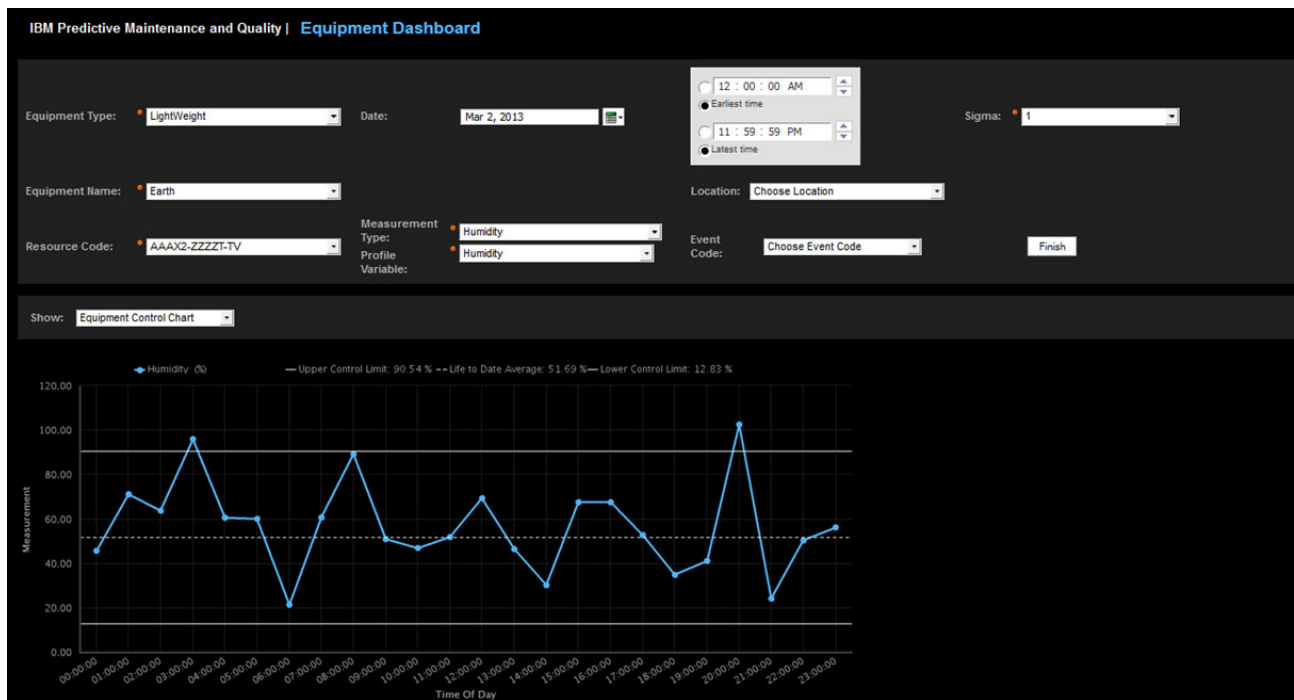


Figure 13-13 Equipment Control Chart for an asset

Equipment Run Chart

The Equipment Run Chart shows the actual value recorded for a measurement type of a piece of monitored equipment during a reported event. The chart is accessed by selecting **Equipment Run Chart** from the **Show** menu.

The Equipment Run Chart depicted in Figure 13-14 shows the actual value that is recorded for humidity.

The filters that can be applied to this report are Resource Sub Type, Resource Name, Resource Code, Location, Event Code, Calendar Date, Start Time, End Time, Measurement Type, Profile Variable, and Sigma Level.



Figure 13-14 Equipment Run Chart for an asset

Equipment Outlier

The Equipment Outlier report (see Figure 13-15) shows detailed information for a piece of monitored equipment in which performance anomalies have been identified. The report is accessed by selecting **Equipment Outlier** from the **Show** menu.

When the actual value for a particular measurement type goes outside the prescribed limits (low or high) as a result of any unfavorable condition, immediate analysis and monitoring is called for. This report allows all equipment anomalies to be tracked in a single place.

The filters that can be applied to this report are Resource Sub Type, Resource Name, Resource Code, Location, and Event Code.

IBM Predictive Maintenance and Quality | **Equipment Dashboard**

Equipment Type:

LightWeight

Sigma:

1

Equipment Name:

Earth

Location:

Choose Location

Resource Code:

AAAX2-ZZZZ1-TV

Measurement Type:

AM Count

Alarm Count

Assembly Time

Breakdown Count

Component Life Test

Component Life Test Count

Component Life Test Defect Count

Component Load Test

Select all Deselect all

Event Code:

Choose Event Code

Finish

Show:

Equipment Outliers

Measurement Type	Life to Date Average	Lower Control Limit	Upper Control Limit	Last Value	Units of Measure
DEV Measurement of Type	1.71	-2.82	6.25	12.00	
FAIL Measurement of Type	1.24	-3.04	5.52	43.16	
Forecasted Days to Next Maintenance	10.43	9.29	11.56	13.00	
Maintenance Health Score	0.32	0.31	0.34	0.36	
Operating Hours	51.42	12.50	90.34	104.48	hrs
RPM	93.74	58.11	129.37	11.14	
Scheduled Days to Next Maintenance	0.14	-0.24	0.52	1.00	

Figure 13-15 Equipment Outlier report

Chapter 13. Additional reports 177

Event Type History

The Event Type History report (see Figure 13-16) displays the recorded value of different measurement types for a piece of monitored equipment during past events. The report is accessed by selecting **Event Type History** from the **Show** menu. The report includes an option to choose a specific target date range for analysis.

The filters that can be applied to this report are Resource Sub Type, Resource Name, Resource Code, Location, Event Code, Calendar Date, and Event Type.

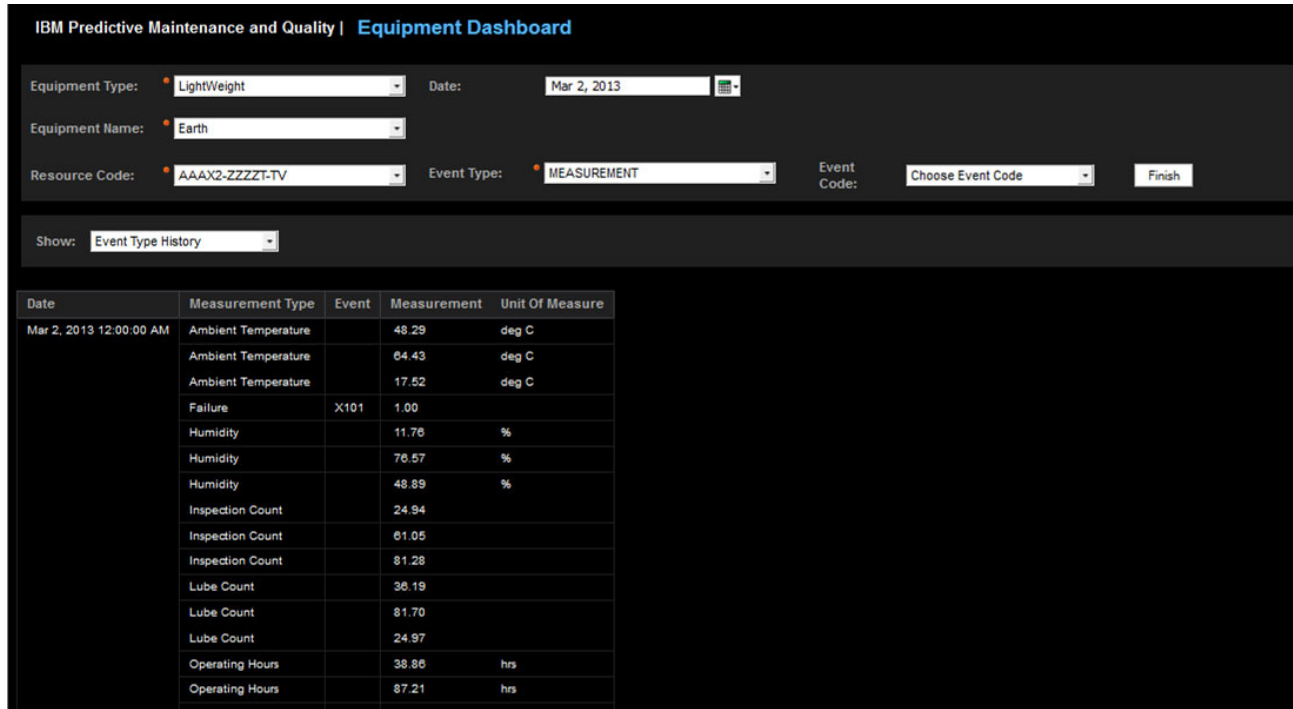


Figure 13-16 Event Type History report

13.2.4 Product Quality Dashboard

The Product Quality Dashboard provides an overview of defects and inspection rates of a monitored asset or its components. The top portion of the dashboard shows the data related to defects and analyses, whereas the bottom portion presents the graphs.

The Product Quality Dashboard includes several subordinate reports and charts. These are explained in the following subsections.

Defect Analysis

The Defect Analysis report (Figure 13-17) uses pie charts to show the distribution of defects. The report is accessed by selecting **Defect Analysis** from the **Show** menu.

By viewing the pie charts, a user can understand where the highest percentage of defects originates. Multiple charts can be displayed to show defect distribution in different dimensions.

The filters that can be applied to this report are Process Hierarchy and From Data and To Date.

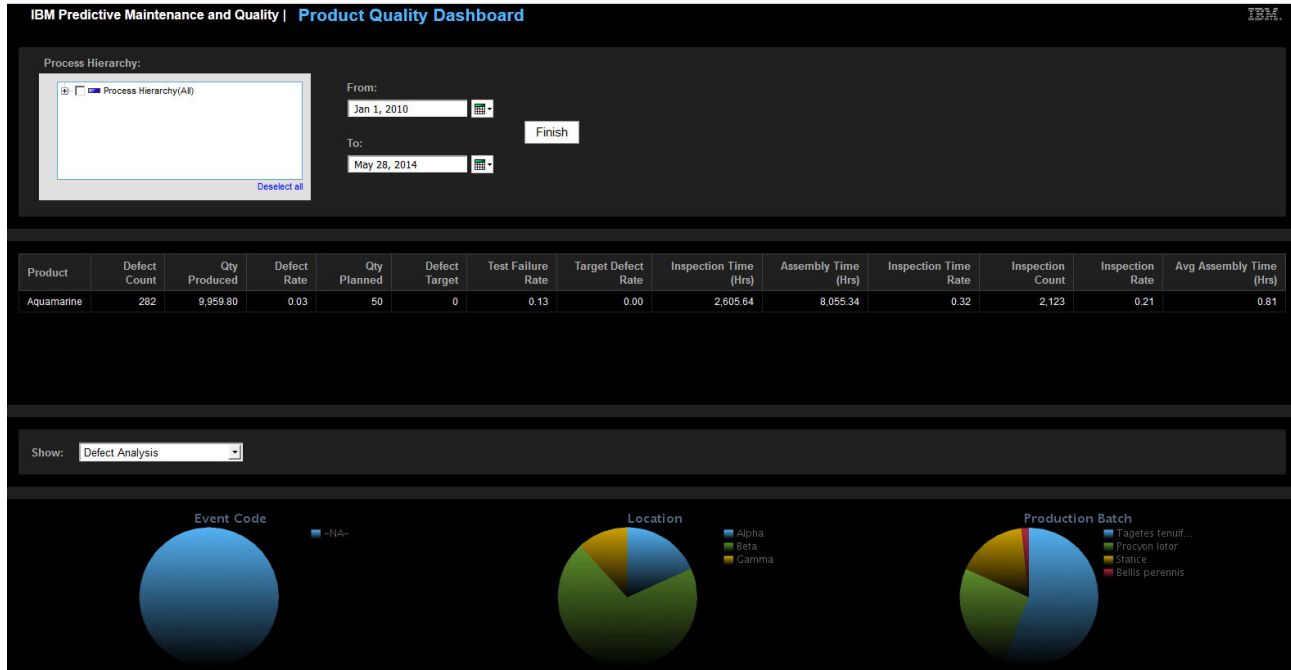


Figure 13-17 Defect Analysis report

Inspection Rate Analysis

The Inspection Rate Analysis report examines the relationship between inspections and defects over time, so you can determine the optimal rate at which inspections should be performed. The report is accessed by selecting **Inspection Rate Analysis** from the **Show** menu.

The report (Figure 13-18) includes a Defect Summary report, an Actual Defect versus Target Defect bar chart, and a Defect Rate versus Inspection Rate line chart.

The filters that can be applied to this report are Process Hierarchy and From Date and To Date.

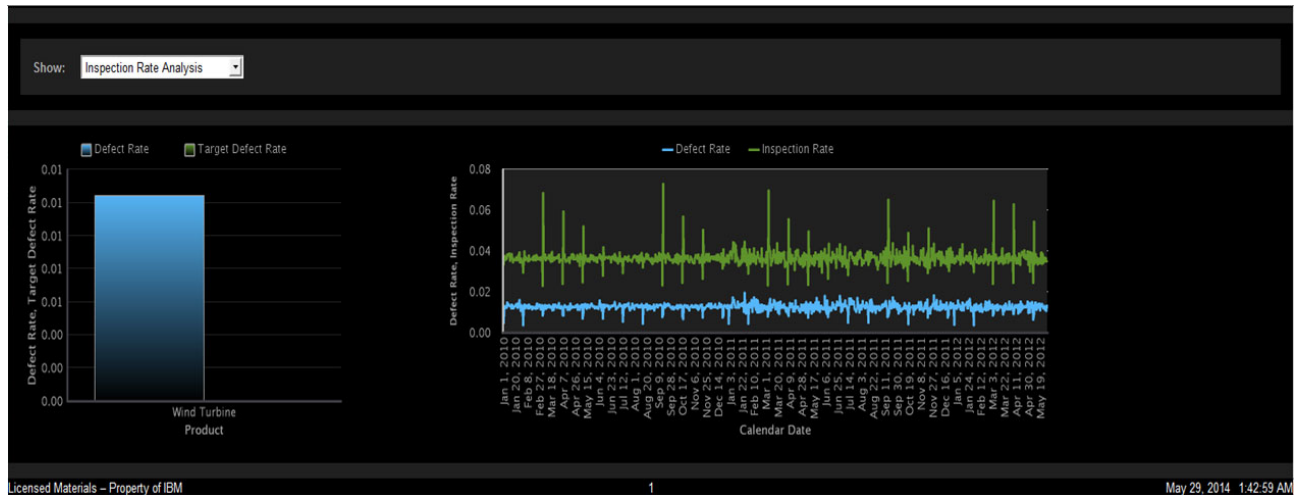


Figure 13-18 Inspection Rate Analysis Report

Material Usage by Process

The Material Usage by Process report (Figure 13-19) provides an overview of material usage, which refers to the number of measurements that are taken for a material in the company's production processes. The calculations are based on the Period Measure Count profile variable, which represents the number of measurements that are taken in a single reporting period (one day, by default). The results are displayed in the Material by Type, Supplier, and Batches by Product hierarchies that are created in the IBM Cognos Framework Manager model.

The filter that can be applied to this report is Process Hierarchy.

Show: Material Usage By Process			
Period Measure Count	<u>MAN Made Material</u>	<u>Material Harness from Nature</u>	<u>Not Applicaable</u>
	Wind Turbine GE US	Wind Turbine GE US	Not Applicable
<u>Assembly</u>	2,486	1,290	1,257
<u>Component Life Test</u>	1,248	600	625
<u>Product Life Test</u>	1,276	607	616
<u>Product Load Test</u>	1,196	644	539
<u>Section Load Test</u>	1,349	640	582

Figure 13-19 Material Usage By Process report

13.2.5 Audit

The Audit report (see Figure 13-20) shows the counts of each row in the solution's primary master data tables. Each count number is a hyperlink that can be used to view pertinent details from the related master data table.

IBM Predictive Maintenance and Quality Audit Report		
Resources	AGENT	4
	ASSET	9
Materials by Type	Manufactured	4
	Natural	3
Processes		10
Batches by Product	Aquamarine	11
Profile Variables		82
Measurement Type		46

Figure 13-20 Audit report

13.2.6 Drill Through Lists

The Drill Through Lists report shows details about assets and associated attributes. These reports can be configured for additional attributes as required.

Note: Drill Through Lists 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 Drill Through Lists reports on their own.

The most popular drill-through lists are described here:

- ▶ Asset List (displayed according to the Resource by Type hierarchy)

This list (Figure 13-21) shows details for available resources, including Resource Type, Resource Sub Type, Resource Name, Resource Code, Model, Location Name, Organization, Manufacturer, and Service Plan.

IBM Predictive Maintenance and Quality ASSET List										
Resource Type Name	Resource Sub Type	Resource Name	Resource Code	Model	Location Name	Broad Peak	Broad Peak	Broad Peak	Broad Peak	Broad Peak
ASSET	LightWeight	Aura	AAW5-ZZZZ-TT	XY	Epsilon	Logan	Logan	Logan	Logan	Logan
	LightWeight	Earth	AAW2-ZZZZ-TV	XY	Beta	Everest	Everest	Everest	Everest	Everest
	LightWeight	Tornado	AAW8-ZZZZ-TA	YTT	Theta	McKinley	McKinley	McKinley	McKinley	McKinley
	Medium Load	Hydra	AAW6-ZZZZ-TR	XY	Zeta	Lhotse	Lhotse	Lhotse	Lhotse	Lhotse
	Medium Load	Lunar	AAW3-ZZZZ-TP	XY	Gamma	McKinley	McKinley	McKinley	McKinley	McKinley
	Power Saver	Solar	AAW1-ZZZZ-TC	XY	Alpha	Lhotse	Lhotse	Lhotse	Lhotse	Lhotse
	Power Saver	Tidal	AAW7-ZZZZ-TW	YTT	Eta	Everest	Everest	Everest	Everest	Everest
	Power Saver	Volcano	AAW4-ZZZZ-TY	YTT	Delta	Broad Peak	Broad Peak	Broad Peak	Broad Peak	Broad Peak
	Unspecified	Not Applicable	-NA-	-NA-	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable

Figure 13-21 Asset List

- ▶ Material List (displayed according to the Material by Type hierarchy)

This list (see Figure 13-22) shows details about available materials, including Material Type, Material Name, and Supplier.

IBM Predictive Maintenance and Quality Material List		
Material Type	Material	Supplier
Natural	Fiberglass	AAE-XQ-AAB-04
	Polymer	WP_SUPP_01
	Rubber	AAC-XZ-AAC-02

Figure 13-22 Material List

- Process List (displayed according to the Process by Type hierarchy)

This list (see Figure 13-23) shows details about available processes, including Process Name and Process Level.

IBM Predictive Maintenance and Quality Process List					
Process	Process Level 1	Process Level 2	Process Level 3	Process Level 4	Process Level 5
Amazon	Amazon	--	--	--	--
Congo	Amazon	Congo	--	--	--
Ganges	Ganges	--	--	--	--
Mekong	Mekong	--	--	--	--
Mississippi	Mekong	Mississippi	--	--	--
Nile	Nil	--	--	--	--
Salmon	Nile	Salmon	--	--	--
Thames	Nile	Salmon	Tamise	--	--
Yangtze	Yangtze	--	--	--	--
Yellow	Yangtze	Amarelo	--	--	--

Figure 13-23 Process List

- Production Batch List (displayed according to the Batches by Type hierarchy)

This list (Figure 13-24) shows details about the available production batch, including Product Name and Production Batch.

IBM Predictive Maintenance and Quality Production Batch List	
Product	Production Batch
Aquamarine	Acacia
	Allium
	Bellis perennis
	Campanula
	Castor
	Dactylorhiza
	Lavandula
	Melospiza lincolnii
	Procyon lotor
	Statice
	Tagetes tenuifolia

Figure 13-24 Production Batch List

► Profile Variable List

This list (see Figure 13-25) includes all measures and key performance indicators that are used for the resources, processes, and materials that are being tracked in daily profiles and historical snapshots.

Profile Variable Name	Profile Measure Name	Profile Variable Code	Profile Calculation Name	Units	Aggregation Type	Comparison String	Low Value Date	High Value Date	Low Value Number	High Value Number	KPI Indicator	Profile Indicator	Data Type	Carry Forward Indicator	Process Indicator	Resource Type	Material Type
AM Count	Actual Maintenance	AMC	Measurement of Type Count		SUM						1	1	INT	0	0	ASSET	Not Applicable
Alarm Count	Alarm Count	AC	Measurement of Type Count		SUM						1	1	INT	0	1	ASSET	Not Applicable
Assembly Time	Assembly Time	ATIME	Measurement of Type	hrs	SUM						1	1	INT	0	1	ASSET	Not Applicable
Breakdown Count	Breakdown	BC	Measurement of Type Count		SUM						1	1	INT	0	1	ASSET	Not Applicable
Component Life Test	Component Life Test	CLT	Measurement of Type		AVERAGE						0	1	INT	0	1	ASSET	Not Applicable
Component Life Test Count	Component Life Test	CLTX	Measurement of Type Count		SUM						1	1	INT	0	1	ASSET	Not Applicable
Component Life Test Defect Count	Component Life Test	CLTXX	Measurement Above Limit		SUM					95.00	1	1	INT	0	1	ASSET	Not Applicable
Component Load Test	Component Load Test	CELLD	Measurement of Type		AVERAGE						0	1	INT	0	1	ASSET	Not Applicable
Component Load Test Count	Component Load Test	CELLDX	Measurement of Type Count		SUM						1	1	INT	0	1	ASSET	Not Applicable
Component Load Test Defect Count	Component Load Test	CELLDXX	Measurement Above Limit		SUM					90.00	1	1	INT	0	1	ASSET	Not Applicable
DEV Measurement of Type	Deviation	DEV	Measurement of Type		SUM						1	1	INT	0	0	ASSET	Not Applicable
FAIL Above Limit	Failure	FAIL Above Limit	Measurement Above Limit		AVERAGE					10.00	1	1		0	0	ASSET	Not Applicable

Figure 13-25 Profile Variable List

► Measurement Type List:

This list (see Figure 13-26) displays the various measurement types that are in use. For each measurement type, the report shows the applicable unit of measure and aggregation type.

Measurement Type Name	Unit Of Measure	Aggregation Type	Carry Forward Indicator	Event Code Indicator
Actual Maintenance		SUM	0	0
Alarm Count		SUM	0	0
Ambient Temperature	deg C	AVERAGE	0	0
Assembly Time	hrs	SUM	0	0
Breakdown		SUM	0	0
Component Life Test		AVERAGE	0	1
Component Load Test		AVERAGE	0	1
Deviation		AVERAGE	0	0
Failure		SUM	0	1
Forecasted Days to Next Maintenance		AVERAGE	0	0
Humidity	%	AVERAGE	0	0
Humidity_Importance		AVERAGE	0	0
INSPECT		SUM	0	0
Inspection Count		SUM	0	0
Inspection Time	hrs	SUM	0	0
Inspection_Importance		AVERAGE	0	0
Integrated Health Score		AVERAGE	0	0
LUBE_Importance		AVERAGE	0	0
Lube Count		SUM	0	0
Maintenance Health Score		AVERAGE	0	0

Figure 13-26 Measurement Type List

13.2.7 Material Usage By Production Batch

The Material Usage By Production Batch report (Figure 13-27) provides an overview of material usage by production batch. By correlating production batches with defects to material usage, you can trace the impact of defective materials.

The report uses Period Measure Count, which is the number of measurements that are taken in one period. The filters that can be applied to this report are Process Hierarchy and Event Code.

IBM Predictive Maintenance and Quality | **Material Usage By Production Batch**

Process Hierarchy:

+

Process Hierarchy(All)

Deselect all

Event Code:

Choose Event Code

Finish

Profile:

Choose Profile

Period Measure Count		<u>Manufactured</u>	<u>Natural</u>
		AAB-XY-AAB-01	WP_SUPP_01
Aquamarine	Procyon lotor		6
	Tagetes tenuifolia	10	

Figure 13-27 Material Usage By Production Batch

13.2.8 Advance KPI Trend Chart

The Advance KPI Trend Chart (Figure 13-28) compares multiple 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, but you can also drill down to a daily chart.

Each chart displays data for one profile and all of the resources that you select. By default, the chart displays all resources and all profiles, but for maximum clarity, you should select just a few related profiles to analyze across a set of resources. To see a month's worth of data separated by day, click a specific data point or click the month on the X axis.

The filters that can be applied to this report are From Date, To Date, Location, Resource Sub Type, Resource, Profiles, and Event Code.

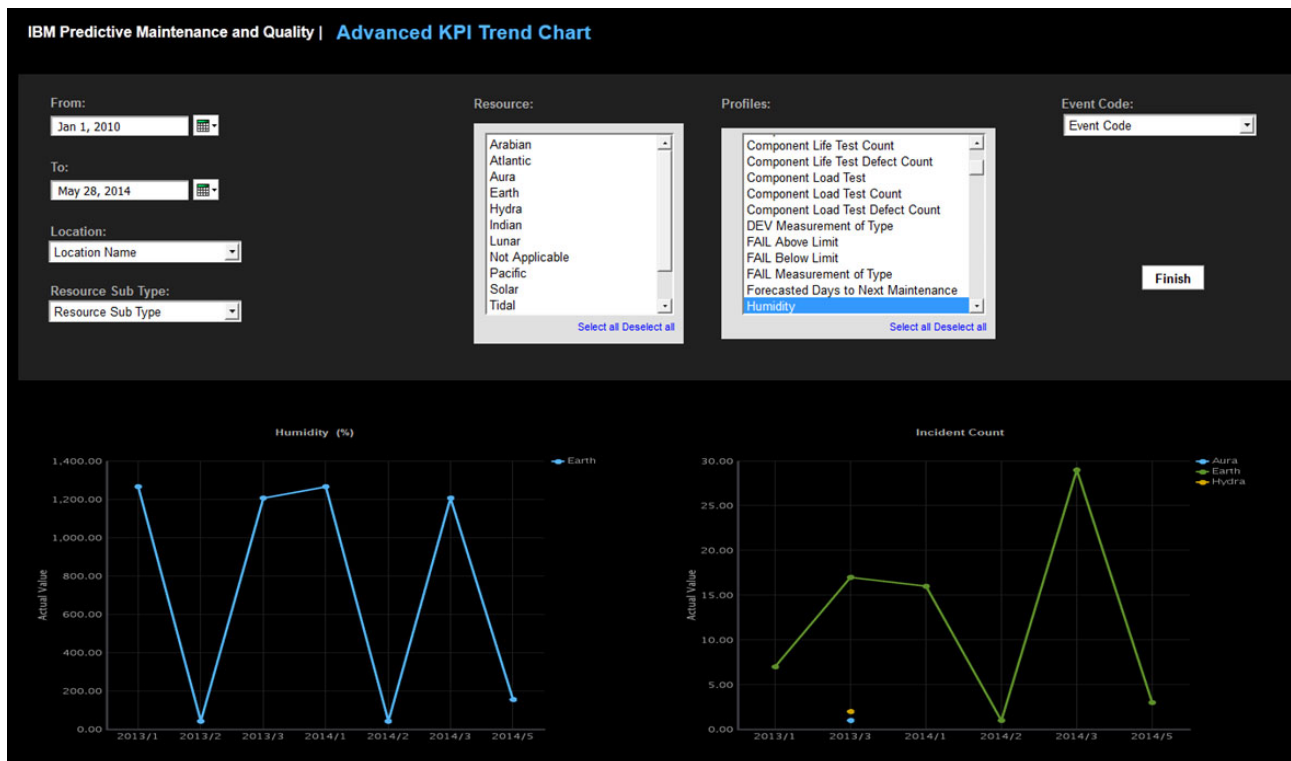


Figure 13-28 Advanced KPI Trend Chart



Execution, training, and deployment of SPSS models

This appendix provides some background on the invocation, execution, training, and deployment options for the IBM SPSS models that are provided in ready-to-use fashion with the IBM Predictive Maintenance and Quality solution. Specifically, it describes (at a high level) the different ways the models are scheduled and run, the deployment option used across all of the models, and the message flows responsible for triggering the model training process.

This appendix includes the following sections:

- ▶ Model execution
- ▶ Model Refresh deployment option
- ▶ Model evaluation
- ▶ Model training invocation

Model execution

SPSS models are developed to run in different modes. Each invocation mode is explained here.

Batch mode

When an SPSS model is developed to run in batch mode, it is scheduled to run on a predefined schedule. Both model building and scoring happens at the same time.

SPSS batch models are hosted as jobs in IBM SPSS Collaboration and Deployment Services (C&DS) and are triggered through IBM Integration Bus message flows by using an SPSS job URI. The job starts the series of SPSS streams shown in Figure A-1:

- ▶ INTEGRATION_HEALTH_DATA_PREPARATION.str
- ▶ INTEGRATION_HEALTH_COMBINED.str
- ▶ INTEGRATION_STATUS_FAIL.str
- ▶ INTEGRATION_STATUS_SUCCESS.str

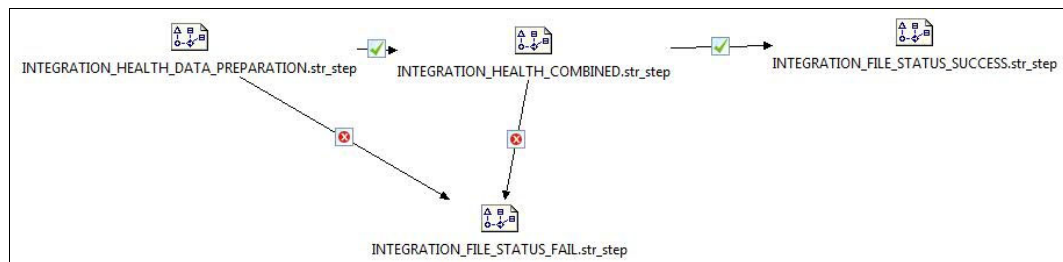


Figure A-1 SPSS job flow

The involved SPSS streams access appropriate database tables. After the data is read, the model is built and its output is written back to the staging tables. The results in the staging tables are then written to a comma-separated values (CSV) file and passed on to appropriate Integration Bus message flows. An extra CSV file is created to log the success or failure of the job. The log file is read by Integration Bus message flows to trigger appropriate responsive action.

Training mode

Training mode is a specialized form of batch mode in which models are trained based on the data available in database, but no scoring is performed. SPSS training streams internally persist the training results, which will eventually be referenced during real-time scoring. SPSS C&DS jobs are used to initiate model training, and these jobs are scheduled and started as explained later in this appendix.

Real-time mode

In real-time mode, SPSS provides the requested predictive results based on the trained model. There is no database interaction in real-time scoring mode. The objective is to perform instant analysis as data (such as measurements or sensor readings) is received. Preparation of the input data is done by using Integration Bus message flows.

Model Refresh deployment option

Model Refresh is the process of rebuilding an existing model in a stream using newer data. The stream itself does not change in the repository. For example, the algorithm type and stream-specific settings remain the same, but the model is retrained on the new data and updated.

To enable automatic refresh of the models, the Model Refresh deployment type is selected (Figure A-2).

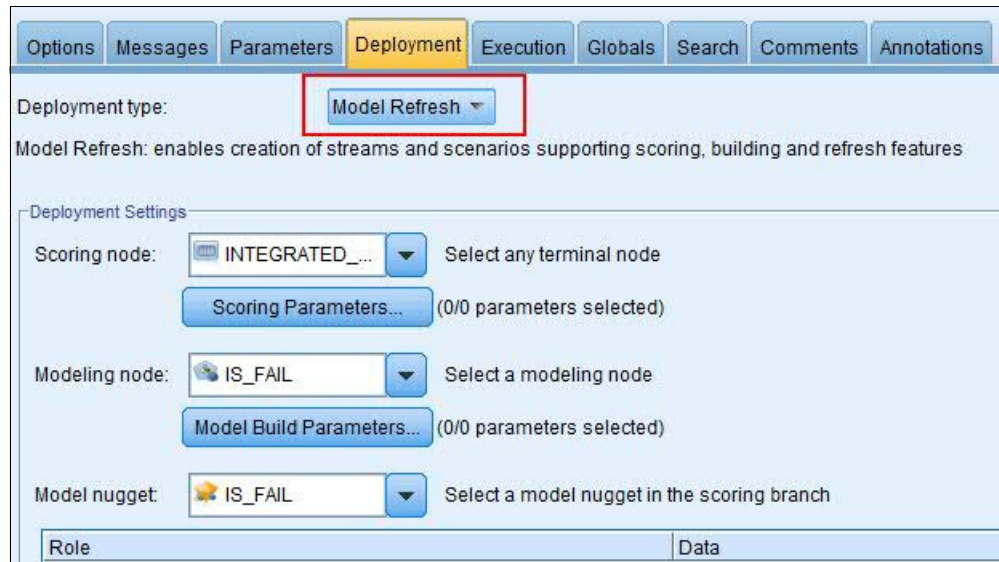


Figure A-2 Stream Deployment tab with Model Refresh option selected

Model evaluation

Model evaluation is a process of verifying the results of the model against known results involving a sample of data from the past. Depending on the problem being studied and the type of analytics performed, this evaluation can involve different techniques and samples of data.

After model evaluation, there are two possible next steps:

- ▶ Continue to the deployment phase: This involves incorporating the model results into the process and producing a final report, if appropriate.
- ▶ Refine or replace the models: If results are found to be almost, but not quite, optimal, another round of modeling must be considered. The object is to take what was learned so far to produce better results in future runs of the models.

Model training invocation

This subsection covers the training of the sensor and integration analytics model. The PMQ solution includes IBM Integration Bus message flows to trigger training of the sensor health score and integrated health score models. These message flows enable an automated way to train the models.

HSTrainingTimer message flow

The HSTrainingTimer message flow (Figure A-3) runs once every 90 days to initiate the training of the sensor model.

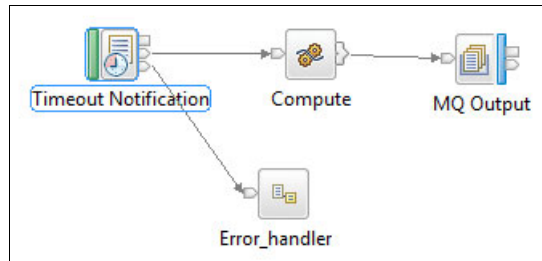


Figure A-3 HSTrainingTimer message flow

TrainingJobIntegration message flow

The TrainingJobIntegration message flow (Figure A-4) triggers the SPSS job to train the sensor and integration analytics models.

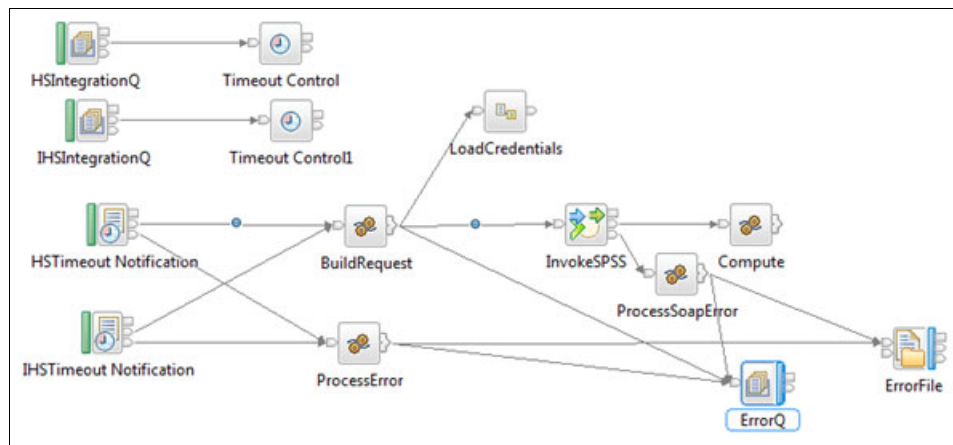


Figure A-4 TrainingJobIntegration message flow

FileIntegration message flow

Upon completing the training of the jobs, SPSS places the Training Eligibility file (which contains the list of resources and their eligibility for sensor health score and integrated health score training, depending on the data in RESOURCE_KPI) and Training Status file (which contains the status of SPSS sensor health score and integrated health score training) in the /integrationin directory. The FileIntegration message flow (Figure A-5) moves these files from the /integrationin directory to a log directory.

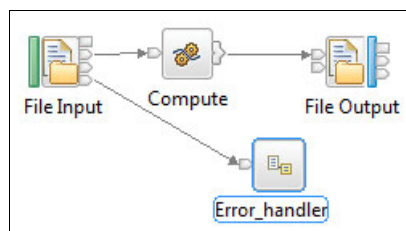


Figure A-5 FileIntegration message flow



REDP-5035-01

ISBN 0738454257

Printed in U.S.A.

Get connected

