Using WebSphere MQ Telemetry and Pachube to connect to remote sensors and devices

Nick Maynard

June 01, 2011

Pachube and the MQTT protocol with its accompanying broker implementations both offer excellent facilities for aggregating data from remote devices and sensor networks into a common API. However, they do not provide a way to integrate data between the two technologies. This article shows you how to use the new WebSphere MQ Telemetry function to bridge the two networks and share data, and includes a downloadable utility to facilitate the integration.

Introduction

IBM® WebSphere® MQ Telemetry extends the universal messaging backbone provided by WebSphere MQ to a wide range of remote sensors, actuators, and telemetry devices. It uses the MQ Telemetry Transport (MQTT) protocol to enable you to monitor enterprise sensors using IBM MQ technology. MQTT is a messaging protocol developed by IBM, previously available in WebSphere Message Broker, and donated by IBM to the open-source community. WebSphere MQ Telemetry includes a client Java™ library to enable the development of MQTT-enabled code.

Another increasingly popular technology for monitoring sensors and devices is Pachube. Pachube and MQTT are enabling technologies for the Internet of Things, and are expected to be serious players in the interconnected, instrumented world envisioned by the IBM Smarter Planet initiative.

Pachube currently has about 120,000 users (according to statshow.com), and is increasingly being used by providers such as CurrentCost to provide sensor interconnectivity. The IBM MQTT protocol is used in many diverse environments to connect a variety of sensor systems.
This article discusses the challenges of integrating the two systems and presents potential solutions, including a utility for integrating the networks. The article should be valuable to users working with the two systems or merely comparing them, and the utility should be invaluable to those integrating the two systems.

## Subscribing and publishing to MQTT topics

The system design of the Pachube-MQTT bridge involves publishing information from Pachube feeds to MQTT topics, and listening on MQTT topics for information to publish to Pachube feeds, as shown below:

![System design of Pachube-MQTT bridge, with mapping of Pachube feeds to MQTT topics and vice-versa](image)

MQTT is designed from the group up to provide a publish/subscribe message model in an easy-to-understand way based on "topics." Because the protocol is openly published with a royalty-free license, many client and broker implementations exist for it. Typically, an individual source of data is mapped to an MQTT topic, which usually means that each source of sensor information will have its own topic.

## Enabling technologies

The new WebSphere MQ Telemetry release includes a simple Java client, and you can use this library to both listen for updates, and to submit updates to topics.

## Subscribing and publishing to Pachube feeds

When bridging MQTT and Pachube, you need to listen to Pachube feeds for information to publish onto MQTT topics, and publish information onto Pachube feeds from MQTT topics.

The Pachube API leverages the HTTP protocol to provide the interface both for publishing feed updates into the system, and for retrieving feed updates from the system. Publishing feed updates automatically is accomplished solely by pushing data over HTTP using PUT. While it is possible to "poll" for updates to feeds, the Pachube API provides a facility for registering an explicit listener, enabling Pachube to "push" updates to an application. Listeners must implement a basic subset of the HTTP protocol to be able to implement this part of the Pachube API.
Enabling technologies

Use the Apache HttpClient library for publishing updates to the Pachube system, and the Jetty lightweight Web server to implement the listener component. Additionally, Pachube pushes feed updates to listeners in a JSON envelope, so you can use the Jackson JSON library to deserialise this structure.

Integrating the networks

Any attempt to bridge the two networks must take into account the differences between the two systems as part of its design.

Throttling traffic

As a protocol, MQTT has no throttling features. However, to reduce traffic on their system, and to provide a differentiator for their premium customers, Pachube implements multiple levels of throttling for their clients. Throttling is particularly apparent in the area of publishing updates to Pachube -- basic site membership allows only three updates per minute. Therefore any integration must handle this publish throttling in an intelligent manner; naive implementations may lose updates, or fail to update feeds that receive updates less frequently. The following algorithm provides a fair solution that balances the need to provide timely updates with the need to minimise loss of information:

1. When an update arrives from an MQTT topic, determine the target feed and add the update to an internal queue structure. If an update already exists in the queue for the target feed, replace it with the new value, but keep the original timestamp.
2. As often as permitted by the Pachube throttling solution (if three updates per minute, then every 20 seconds), take the update with the oldest timestamp from the internal queue structure, publish it to Pachube, and remove it from the internal queue structure.

This algorithm is not perfect and certain edge cases exist. However for the majority of low-medium throughput scenarios, it should suffice. The basic utility for integration, introduced below, uses a similar algorithm.

Authentication and authorisation

Both technologies provide authentication facilities, but levels of authorisation vary.

The MQTT protocol lets brokers optionally control access using username and password pairs, but once authenticated, all actions are authorised.

Pachube allows all current data to be read by any user using the Web interface. For automatic implementations, authentication is governed using API keys. All API keys are authorised to register listeners for any feed, and API keys are authorised to publish updates to feeds by the feed owner.

A bridging utility must therefore at a minimum enable you to configure the Pachube API key on a per-feed basis.
Using the basic utility for integration

This article includes a basic utility that uses the principles described above to provide a simple integration bridge between the Pachube network and the MQTT protocol. You can download it below, extract it from the zip file, and then use the following instructions to install and configure it:

1. Installation: Download the JAR files listed in the file lib/README.txt to run the program.
2. Configuration: Copy config.xml.dist to config.xml, and edit it to suit your requirements.
3. Execution: Run bridge.bat or bridge.sh according to your system type.
4. Debugging: Edit the log4j.properties file to set per-package debugging levels.

Conclusion

This article showed you how to use the new WebSphere MQ Telemetry function to bridge the Pachube and MQTT networks and share data between them. It also introduced a basic utility to bridge the two networks.
## Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code sample</td>
<td>mqttPachubeBridge.zip</td>
<td>30KB</td>
</tr>
</tbody>
</table>
Related topics

- **WebSphere MQ and related resources**
  - **MQ Telemetry Transport (MQTT) protocol**
    The MQTT protocol was created by IBM and donated to the open-source community, and it is described on the independent mqtt.org Web site.
  - **WebSphere MQ Telemetry**
    The Telemetry feature is available in WebSphere MQ V7.0.1 or later, and provides full support for the MQTT V3.1 protocol as well as interoperability with WebSphere MQ.
  - **Pachube**
    Visit the Pachube Web site and explore the feeds available.
  - **The Internet of Things**
    Comments by Juniper Networks Social Business Strategist Adam Christensen at asmatterplanet.com.
  - **WebSphere MQ developer resources page**
    Technical resources to help you design, develop, and deploy messaging middleware with WebSphere MQ to integrate applications, Web services, and transactions on almost any platform.
  - **WebSphere MQ product page**
    Product descriptions, product news, training information, support information, and more.
  - **WebSphere MQ documentation library**
    WebSphere MQ product manuals.
  - **WebSphere MQ V7 information center**
    A single Web portal to all WebSphere MQ V7 documentation, with conceptual, task, and reference information on installing, configuring, and using WebSphere MQ V7.
  - **IBM Redbook: WebSphere MQ V7 features and enhancements**
    Describes the fundamental concepts and benefits of message queuing technology, describes the new features in V7, and provides a business scenario that shows those features in action.
  - **Download a free trial version of WebSphere MQ V7**
    A 90-day, full featured, no-charge trial of WebSphere MQ V7
  - **WebSphere MQ support page**
    A searchable database of support problems and their solutions, plus downloads, fixes, and problem tracking.
  - **WebSphere MQ public newsgroup**
    A non-IBM forum where you can get answers to your WebSphere MQ technical questions and share your WebSphere MQ knowledge with other users.
  - **WebSphere MQ SupportPacs**
    Downloadable code, documentation, and performance reports for the WebSphere MQ family of products.
- **WebSphere resources**
  - **developerWorks WebSphere developer resources**
    Technical information and resources for developers who use WebSphere products. developerWorks WebSphere provides product downloads, how-to information, support
resources, and a free technical library of more than 2000 technical articles, tutorials, best practices, IBM Redbooks, and online product manuals.

- **developerWorks WebSphere application connectivity developer resources**
  How-to articles, downloads, tutorials, education, product info, and other resources to help you build WebSphere application connectivity and business integration solutions.

- **developerWorks WebSphere business process management developer resources**
  WebSphere BPM how-to articles, downloads, tutorials, education, product info, and other resources to help you model, assemble, deploy, and manage business processes.

- **developerWorks WebSphere SOA and Web services developer resources**
  How-to articles, downloads, tutorials, education, product info, and other resources to help you design and build WebSphere SOA and Web services solutions.

- **Most popular WebSphere trial downloads**
  No-charge trial downloads for key WebSphere products.

- **WebSphere forums**
  Product-specific forums where you can get answers to your technical questions and share your expertise with other WebSphere users.

- **WebSphere on-demand demos**
  Download and watch these self-running demos, and learn how WebSphere products and technologies can help your company respond to the rapidly changing and increasingly complex business environment.

- **developerWorks WebSphere weekly newsletter**
  The developerWorks newsletter gives you the latest articles and information only on those topics that interest you. In addition to WebSphere, you can select from Java, Linux, Open source, Rational, SOA, Web services, and other topics. Subscribe now and design your custom mailing.

- **WebSphere-related books from IBM Press**
  Convenient online ordering through Barnes & Noble.

- **WebSphere-related events**
  Conferences, trade shows, Webcasts, and other events around the world of interest to WebSphere developers.

- **developerWorks resources**
  - **Trial downloads for IBM software products**
    No-charge trial downloads for selected IBM® DB2®, Lotus®, Rational®, Tivoli®, and WebSphere® products.
  - **developerWorks blogs**
    Join a conversation with developerWorks users and authors, and IBM editors and developers.
  - **developerWorks tech briefings**
    Free technical sessions by IBM experts to accelerate your learning curve and help you succeed in your most difficult software projects. Sessions range from one-hour virtual briefings to half-day and full-day live sessions in cities worldwide.
  - **developerWorks podcasts**
    Listen to interesting and offbeat interviews and discussions with software innovators.
  - **IBM Education Assistant**
A collection of multimedia educational modules that will help you better understand IBM software products and use them more effectively to meet your business requirements.

© Copyright IBM Corporation 2011
Trademarks
(www.ibm.com/developerworks.ibm/trademarks/)