WebSphere software

IBM

# Integrating monitoring and telemetry devices as part of enterprise information resources.

WebSphere MQ Integrator

March 2002

*By Andy Stanford-Clark,*
*WebSphere MQ Development,*
*IBM Software Group*

e business software

Contents



## Introduction

The mantra for the enterprise today is end-to-end business integration. If you're responsible for information technology (IT) systems, or you depend heavily on the service your IT systems provide, you're acutely aware of the importance of connecting data and applications throughout the enterprise. Enterprise integration means processes can be automated, cycle times can be reduced and resources can yield their actual value.

In many industries, data (like oil flow rates, for instance) from remote monitoring and telemetry devices is becoming increasingly valuable through the enterprise as a whole, and not just as raw data for front-end applications.

The problem is that this valuable data is qualitatively different from the data that's generated within the enterprise, because of the physical constraints on collecting it and delivering it for processing. The result is that the data tends to remain at the edges of the enterprise, processed by dedicated applications that act as a data barrier to the rest of the IT systems.

There is a solution, unique to the IBM MQSeries® family of integration products, that enables you to treat the valuable resource of field data in exactly the same way as other enterprise data, and make it freely available throughout your enterprise.

MQSeries products transform field data into whatever form is required by different applications, and distribute it to all applications that have registered an interest in that data topic. The flexibility of this method means that new ways of leveraging data can be implemented speedily, without extensive reprogramming or systems administration.

The result is integrated data and applications that reflect accurately the full extent of your enterprise. And your managers can take full advantage of their information resources.

This white paper describes a system, based on a publish/subscribe broker, that manages the flow of information from remote devices to any enterprise applications that need the data (and also from command and control applications out to remote devices in the field). The system is based on the IBM MQSeries family of products (now known as WebSphere MQ) and, specifically, on a new publish/subscribe protocol, MQ Integrator SCADA device protocol (MQIsdp) which operates through the family's publish/subscribe broker, WebSphere MQ Integrator.

Telemetry data comes from a variety of remote monitoring stations, such as oil and gas pipeline sensors, transmitted using TCP/IP over telecommunications links, ranging from dial-up phone connection to satellite link. The data reaches the broker within the enterprise, where the powerful message transformation capabilities of WebSphere MQ Integrator transform the message from (typically) compressed binary representation to a form that is more useful to enterprise applications, such as an XML document.

Using the integration capabilities of the MQSeries family, the data can be made available to any application in your enterprise that can use it, like enterprise resource planning (ERP), billing, scheduling, inventory or trading floors. This is a great step forward because, in most monitoring and telemetry systems, data is available to just one application. Previously, to escape this data cage, IT personnel often resorted to rekeying the data or using complex transformation scripts and proprietary file transfer protocols. The power of the publish/subscribe model is in its ability to allow any application to receive the exact subset of data that it requires, wherever it is in the enterprise.

### Survey of telemetry integration technology

The term telemetry integration is used to cover all aspects of a telemetry and control system, or in the specific sense of a SCADA (Supervisory, Control, And Data Acquisition) system that employs remote monitoring systems for operational purposes. In this paper, the term telemetry integration covers all these – anything to do with remote monitoring, or telemetry and control, is included.

This technology is widespread. Oil and gas pipeline companies are major users, but there are applications in many industries. Energy utilities, for example, are interested in making energy consumption information from meters available to the energy trading floors. With today's systems, traders often work with historical data, typically 30 to 60 days old, so they are never certain of their production, consumption or storage levels. With up-to-the-second information, traders can achieve the very best price for energy nomination trading, leading to greater profits for the energy companies and greater marketplace stability.

Supply chain integration (SCI) and supply chain management (SCM) are other fields with great potential for using telemetry integration, particularly those with energy supply challenges. Because energy companies are now able to monitor power consumption in domestic meters, increases in demand can trigger startup of campus generators, small, gas-powered turbines each generating a few megawatts of power. In turn, this increases the demand for gas, which can be passed back up the supply chain to the gas wellheads, and gas production can be automatically increased.

You can also find this technology in factory automation, security applications, smart building management, weather and environmental monitoring, tracking the location of trucks and railroad cars and so on. In fact, telemetry integration technology can be used to monitor anything that moves, whether it's oil through a pipeline, wind through a weather vane or money into a slot machine.

There's no doubt that data carried by the MQ SCADA protocol has huge potential for exploitation in the enterprise. The problem has been to make the data available — most data-gathering devices are not capable of communicating through MQSeries protocols, or even of using TCP/IP. Instead, they may use proprietary RS232-based serial communications protocols, an analog voltage to be measured, or simply a digital open/close from a relay or switch contact. Today, products are available to connect these simple physical devices to a network.

Industrial network gateway

*Arcom Control Systems (www.arcomcontrols.com), a Spectris company and an IBM Advanced Business Partner based in Kansas City, Kansas and Cambridge, England, specializes in building industrial network gateways, which are small, rugged, embedded computers, running a realtime operating system. The Arcom Director range of products acts as the interface between the proprietary protocols that the physical measurement devices use in the field, and the networked TCP/IP world. The different models in the range come with a variety of input and output (I/O) devices, from analog and digital inputs, solenoid, motor and relay controller outputs and serial I/O.*

*On the network side, the Arcom unit implements a TCP/IP stack and MQIsdp. The unit can communicate over any communication link that supports TCP/IP, including 10baseT Ethernet, an RS232-connected modem using Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP), spread-spectrum radios and satellite networks.*

*The Arcom unit polls locally attached devices, using their native protocols, and runs report by exception (RBE) logic to determine if an event should be reported to the broker. If necessary, a TCP/IP communication link is established, and the MQIsdp protocol driver publishes a message over a TCP/IP socket connection to the broker. The broker can transform the message before routing it into the enterprise.*

*The Arcom unit can also subscribe to topics and receive messages. Thus, control applications can send commands from applications inside the enterprise to the field devices, to open and close valves and switches, start and stop motors and send new instructions to smart devices.*

**Overview of the publish/subscribe model**

The publish/subscribe model is built around a central broker and a number of clients which connect to the broker. The broker acts as a go-between: an agent that matches subscribers to information with publishers of information that's relevant to them. Clients can be publishers of, and/or subscribers to, data and can range from big enterprise-based servers to hand-held pervasive computing devices, such as Palm handhelds, or unattended remote telemetry devices.

Publishers send units of information (called messages) to the broker, on a specific topic. The topic is like the subject line of an e-mail. It tells you what the body of the message contains. Subscribers register their interests in certain topics with the broker.

The broker manages connections from publishers and subscribers, and deals with authentication and access control lists, to control who is allowed to publish and subscribe to which topics. This helps ensure that an accounts department, for example, can't accidentally shut down a pipeline. The broker also transforms messages into a more useful format, like an XML document, and can apply business logic to route messages based on their topics or contents.

It's important to remember that publishers and subscribers are usually unknown to each other—the broker acts as a matchmaker. This decoupling makes the system future-proof, because it generalizes the use of data; it's not tied to specific applications. At any time, you can deploy new applications that can use a new combination of topics. It is this potential that makes publish/subscribe such a powerful concept.

Topics are arranged hierarchically, with slashes (/) between the levels, similar to a URL. The hierarchy, which defines an information space, is carefully designed to help ensure that data is available in a sensible, logical, structure. Often this is shaped by the first subscriber applications to be deployed, but not necessarily. It's difficult to make major changes to the topic hierarchy after the fact, and information architects can give you guidance in this important matter.

Typical application scenario

*Farmers walk their fields with a Global System for Mobile Communications (GSM)-connected Palm Pilot, with a map of the fields. A farmer notices that some crops are drought-stressed and clicks on the appropriate part of the map, selecting Irrigation ON. The PDA sends a message over the GSM modem link using IBM MQSeries Everyplace™. At the WebSphere MQ Integrator broker, the message is transformed into a command message, and sent out using the MQ SCADA protocol, MQIsdp, to a remote device that activates a pump. The sprinkler system starts watering the field before the farmer has closed the PDA.*

When a publisher sends a message, the topic specifies where it belongs in the information hierarchy. Subscribers can either list topics specifically, or use wildcards. For example, a subscriber could specify the maximum weather temperature from London, or from all weather stations.

The broker can add value to the message. Message content may simply be treated as an opaque payload but, for some applications, the data format determines whether the information can be used or not. For example, because of the limited capabilities of remote devices or slow communication links, a flow meter may send compressed data that's intelligible only to specific applications. If the data is in an XML document, it has more utility for most applications, and the broker can transform the data into this—or any other—useful format.

The broker can be thought of as a multiprotocol switching hub, like a network router, but at a higher level in the communications stack. Messages can flow in on one protocol and flow out on a different protocol. In a typical telemetry integration scenario, data flows into the broker using the MQ SCADA protocol, MQIsdp, and from the broker to the enterprise using MQSeries messaging. This gives access to the full range of systems, from mainframes to hand-held devices or personal digital assistants (PDAs), such as a Palm or Psion handheld.

### Evolution of SCADA and telemetry in industry

Despite a huge legacy of installed equipment, the SCADA and telemetry industries are undergoing change, to handle increasing volumes of data, and to make the data more readily available through the enterprise.

With the arrival of more intelligent remote devices, there is a move from a poll/response model to a report by exception (RBE) model. In the poll/response model, a central host contacts each remote device in turn, asks it for its current value, waits for the response and moves to the next device, one after the other, continuously. The time it takes to poll all the devices dictates the minimum period between data samples. This approach is inherently limited in its scalability: The network is used at 100 percent capacity, and running the poll/response protocols on the central host is CPU-intensive.

Example of end-to-end business integration
*A remote SCADA client publishes a message using the MQ SCADA protocol (MQIsdp), reporting that a consignment of oil has been transferred to a customer's tanker. WebSphere MQ Integrator transforms the message into SAP IDoc format, and routes it to the SAP R/3 ERP system to trigger sending a bill to the customer.*

By locating enough intelligence in a remote gateway device to poll locally attached sensors and to implement RBE processing, network traffic can be halved at least (because half of the poll/response conversation is eliminated), but is usually reduced by much more. For example, a temperature might only be reported when it has changed by a whole degree. Similarly, the load is reduced on the host system, which merely waits for values to be reported from the field, so many more field devices can be monitored. The RBE model with a topic name is a classic publish/subscribe application.

Another big change in the industry is the recognition that physical, real-world events can have significant monetary value, particularly on the utility trading floors. It's this recognition that's driving the integration of SCADA and telemetry systems with manufacturing resource planning (MRP), enterprise resource planning (ERP) and production systems. The increased use of industry-standard protocols, and the end-to-end integration that's now expected, highlights the nonstandard nature of proprietary protocols found in the remote telemetry world. Together with the trend to acquisitions and mergers, joining incompatible telemetry systems would create huge challenges, were it not for WebSphere MQ Integrator.

### WebSphere MQ Integrator

The MQSeries family of products has three major components: MQSeries messaging, WebSphere MQ Integrator and MQSeries Workflow. For more details about the MQSeries product family, visit **ibm.com**/mqseries/.

WebSphere MQ messaging is the industry-accepted standard for enterprise application integration (EAI). With more than 7,000 messaging middleware customers, WebSphere MQ is trusted by many of the world's banks and financial institutions to reliably transfer information. WebSphere MQ offers once-and-once-only assured delivery of messages, using an asynchronous queue-based model. An application can send a message to another application and then carry on with other work, even if the other application is not immediately able to process the message. Client applications can put and get messages to and from queues, for point-to-point messaging, or can use Java Message Service (JMS) or the Application Messaging Interface (AMI) for one-to-many publish/subscribe applications.

WebSphere MQ Integrator oversees WebSphere MQ messaging, and some other messaging protocols, and provides the ability to add value to messages as they flow through the system by applying business rules to determine actions or message routing based on the content of the messages. It also has a built-in data dictionary that stores templates for enterprise message formats, so it can perform transformations from one message format to another, including C structures, EDI messages, COBOL copy books and XML documents.

MQSeries Workflow builds on the capabilities of MQSeries messaging and WebSphere MQ Integrator to provide a sophisticated business process modeling capability. Using a simple, graphical modeling interface, business professionals integrate applications and processes to create new processes that extend through the value chain.

#### WebSphere MQ Integrator architecture

WebSphere MQ Integrator handles multiple protocols in and out, including MQSeries messaging, MQIsdp (SCADA protocol) and MQSeries Everyplace (for hand-held pervasive devices, such as Palm Pilot). It has a publish/subscribe engine, providing topic-based and content-based publish and subscribe capability, and managed access control lists to prevent unauthorized publishing or subscribing to topics.

WebSphere MQ Integrator is created around a message flow engine, which flows a message through a series of nodes, each of which operates on the message in a specific way. A nontechnical business analyst using a graphical interface can design the sequence of nodes. The supplied set of standard nodes includes the ability to:

- *Transform messages.*
- *Compute values from input message content.*
- *Publish data to subscribers as incoming topics of interest are received.*
- *Establish Open Database Connectivity (ODBC) connections to Structured Query Language (SQL) databases to retrieve data, which can be used to enrich the content of the message, or to insert or update rows in a database.*
- *Implement business logic, such as making an if-then decision and following two different paths through the message flow based on the outcome of the decision.*

How the MQ SCADA node works

*The node has a TCP/IP socket listener on a configurable socket. The default port is 1883, which has been allocated by the Internet Assigned Numbers Authority (IANA) for the MQ SCADA protocol. The input node supports connections from many devices concurrently, and a message from an MQIsdp client (an Arcom Director unit, for example) starts the message flow to which the SCADA input node is connected. The message propagates through the message flow as usual. Clients can be publishers and/or subscribers, using a single socket connection to the broker.*

The MQ SCADA protocol node is included in WebSphere MQ Integrator, Version 2.02 and higher. This allows remote devices to connect to the broker using the MQIsdp protocol.

**MQ SCADA device protocol – MQIsdp**

MQIsdp is the MQ SCADA device protocol. It is a protocol for publish/subscribe over TCP/IP, with different levels of assurance of delivery. The protocol was specifically designed for the specialized client devices and network types found in telemetry integration applications.

Two key design points of MQIsdp are:

- *The protocol minimizes use of network bandwidth. It has a header overhead of only two bytes. This is of great importance when bandwidth is limited or expensive.*
- *The protocol is simple to implement in embedded systems, and was designed for use in the flow meters and sensors that generate the raw data, as well as industrial network gateways.*

The MQIsdp protocol specification is deliberately nonprescriptive regarding the application programming interface (API) presented to applications on the client device. IBM doesn't supply any client implementations – device and equipment manufacturers implement the protocol on their particular hardware and software combinations. IBM can offer assistance with implementing the protocol, and can supply sample implementations in C, Java™ and Perl for experimentation, education and examples. Arcom Control Systems also offers an integration service for companies that wish to support the protocol in existing installations or within proprietary instruments.

The protocol has a very basic publish/subscribe verb set: connect, disconnect, publish, subscribe, unsubscribe; and an application-level keepalive: pingrequest and pingresponse. Message acknowledgment verbs are used to manage the assured message delivery.

MQIsdp is an open protocol jointly developed by IBM and Arcom Control Systems. The specification is published in the Programming Reference manual for WebSphere MQ Integrator, Version 2.02 and higher (available at **ibm.com**/mqseries/library) so that all interested device manufacturers can implement it.

Two important features of the protocol are assurance of delivery and detection of client disconnection:

Assurance of delivery. MQIsdp offers three levels of delivery assurance, or qualities of service, referred to as QoS0, QoS1 and QoS2. Different qualities of service are appropriate for different types of data, and can be selected on a per-message basis by a publishing application.

- *QoS0 is a simple fire and forget model, which offers at most once delivery. This is similar to the MQSeries nonpersistent messaging mode.*
- *QoS1 uses a simple acknowledgment to provide at least once delivery, which helps ensure, in the event of network failure, a message is eventually delivered. However, loss of the acknowledgment message can result in the client application (correctly) sending the message more than once.*
- *QoS2 has more message flows in the conversation between the client and the broker, but assures a once-and-once-only delivery of the message, compatible with the MQSeries persistent messaging model.*

Detection of client disconnection. This mechanism is used to detect unexpected disconnection, or failure, of the network or the client hardware. When an MQIsdp client connects to the broker, it can optionally specify a special message and topic, and a keepalive time, specified in seconds. If the client fails to publish anything to the broker during the keepalive time, the broker assumes the client was unexpectedly disconnected, and closes the client connection. The broker then publishes the special message using the specified topic on behalf of the client.

### Selected applications and early customer projects

The following are some applications and early customer projects where companies are experiencing benefits of end-to-end integration.

#### Weather station example

The project utilizes an electronic weather station that has sensors providing data on wind speed and direction, rainfall, temperature, humidity and barometric pressure. It has an RS232 output socket, through which it transmits binary-coded data on weather conditions every five seconds.

The weather station is connected to a personal computer, which parses the data and publishes it to a WebSphere MQ Integrator broker using a Perl-language implementation of MQIsdp. The topic hierarchy separates the data into different parameters—such as temperature and wind—and includes the station identifier, so that subscribers can receive data from one particular weather station. Or, by using wildcards, for example, subscribers could receive maximum temperature data from all stations. The broker delivers the data to any connected subscribers, who use an animated Java applet to display the current weather conditions.

#### Electronic Flow Measurement (EFM) and SCADA Operations Project

This project, now live and in production, is an example of the application of this technology to gas or liquid pipeline monitoring control. It's emerging as one of the major implementation scenarios for the WebSphere MQ Integrator telemetry integration technology.

The customer, a large oil and gas pipeline company, uses the VALMET subscriber application (from Metso Automation) for monitoring and controlling its pipelines. The VALMET application runs on a large IBM @server pSeries™, a UNIX® technology-based server, and originally polled approximately 4,000 field devices over a 2Mbit TCP/IP network. Network usage was 100 percent, and there was little spare CPU capacity on the host, preventing the solution from scaling further.

This communication system has been replaced with a WebSphere MQ infrastructure. The various pipeline sensors are polled using Arcom Director units out in the field. The devices communicate with the Arcom unit using 20-mile line-of-sight, spread-spectrum wireless links from Data-Linc Group. When RBE logic running in the Arcom gateway determines that changes in the data must be reported, the Arcom unit connects to the WebSphere MQ Integrator broker using MQIsdp over a TCP/IP network, and publishes the new data values. The broker transforms the messages to provide standardization of formats, and delivers them to VALMET. The VALMET polling back-end has been replaced by an MQ Application Messaging Interface (AMI) application that waits for messages to arrive in an MQ queue.

The network bandwidth is greatly reduced, and the VALMET systems are running at lower CPU usage. In this project, the number of monitoring devices could be tripled, but could be scaled even higher with additional pipeline space.

The SCADA data, including pressure, temperature, flow rate, energy rate and battery voltage, is reported by each station every three to five minutes. An alarm system generates alert messages, which are routed to pagers and e-mail to notify people when exceptional events occur. A data warehouse logs a substantial amount of data that must be held for auditing purposes.

### Automated Meter Reading (AMR) Project

This is a good example of how cumbersome manual systems can give much more value when automated. The original manual system involved faxing data from each of 120 pipeline pumping stations, each with a flow computer which printed out a flow ticket each time the end of a batch of oil passed by. At company headquarters, the faxes were keyed into an Oracle database, so the data could be used by a variety of sophisticated applications within the enterprise.

The new WebSphere MQ infrastructure replaces the manual reporting system with Arcom Director units, each connected by a serial link to a flow computer, communicating through Very Small Aperture Terminal (VSAT) satellite links to the company headquarters. Using MQIsdp, the Arcom Director units publish a standardized electronic flow ticket data representation, to avoid any need for knowledge of device type by the business applications. Using the database insert node to make an ODBC connection to the Oracle database, WebSphere MQ Integrator inserts the data in the same format as the original manual system. The business applications are completely unchanged but now the data appears in the database less than a second after the flow computer generates its flow ticket. Previously it was seven to eight days, depending on how fast the meter reader reached the stations.

#### For more information

For more information about WebSphere MQ Integrator, visit **ibm.com**/mqseries/integrator.

For more information about telemetry integration, visit **ibm.com**/mqseries/integrator/telemetry.



IBM United Kingdom Limited
Hursley Park, Winchester
Hampshire SO21 2JN
United Kingdom

Produced in the United States of America
03-02


G325-5526-00