19

MQ update

January 2001

In this issue

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38126
From USA: 01144 1635 38126
Fax: 01635 38345
E-mail: info@xephon.com

North American office

Xephon/QNA
Post Office Box 350100
Westminster CO 80035-0100, USA
Telephone: (303) 410 9344
Fax: (303) 438 0290

Contributions

Articles published in *MQ Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. For more information about contributing an article you can download a copy of our *Notes for Contributors* from www.xephon.com/contnote.html.

MQ Update on-line

Code from *MQ Update* is available from Xephon's Web site at www.xephon.com/mqupdate.html (you'll need the user-id shown on your address label to access it). If you've a problem with your user-id or password call Xephon's subscription department on +44 1635 33886.

Commissioning Editor

Peter Toogood
E-mail: PeterT@xephon.net

Managing Editor

Madeleine Hudson
E-mail: MadeleineH@xephon.com

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, scripts, and other contents of this journal before making any use of it.

Subscriptions and back-issues

A year's subscription to *MQ Update*, comprising twelve monthly issues, costs £255.00 in the UK; $380.00 in the USA and Canada; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the July 1999 issue, are available separately to subscribers for £22.50 ($33.50) each including postage.

Printed in England.

APPC support using the ‘SideInfo’ dataset (part 2)

This month’s instalment concludes this article on the SideInfo dataset (the first part appeared in last month’s issue of *MQ Update*).

AN EXAMPLE OF A CUSTOMIZED CSQ4SIDE JOB (CONTINUED)

```
SIDELETE
          DESTNAME(SF67809C)
     SIADD
          DESTNAME(SF67809C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67809C)
     SIDELETE
          DESTNAME(SF67740C)
     SIADD
          DESTNAME(SF67740C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67740C)
     SIDELETE
          DESTNAME(SF67755C)
     SIADD
          DESTNAME(SF67755C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67755C)
     SIDELETE
          DESTNAME(SF67759C)
     SIADD
          DESTNAME(SF67759C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67759C)
     SIDELETE
          DESTNAME(SF61041C)
     SIADD
          DESTNAME(SF61041C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF61041C)
     SIDELETE
          DESTNAME(SF67112C)
     SIADD
          DESTNAME(SF67112C)
```

```
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67112C)
SIDELETE
     DESTNAME(SF67Ø48C)
SIADD
     DESTNAME(SF67Ø48C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67Ø48C)
SIDELETE
     DESTNAME(SF67246C)
SIADD
     DESTNAME(SF67246C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67246C)
SIDELETE
     DESTNAME(SF61136C)
SIADD
     DESTNAME(SF61136C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF61136C)
SIDELETE
     DESTNAME(SF67762C)
SIADD
     DESTNAME(SF67762C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67762C)
SIDELETE
     DESTNAME(SF67737C)
SIADD
     DESTNAME(SF67737C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67737C)
SIDELETE
     DESTNAME(SF67652C)
SIADD
     DESTNAME(SF67652C)
     TPNAME(RECV)
     MODENAME(LU62C)
     PARTNER_LU(SF67652C)
SIDELETE
     DESTNAME(SF61Ø66C)
SIADD
     DESTNAME(SF61Ø66C)
     TPNAME(RECV)
```

```
          MODENAME(LU62C)
          PARTNER_LU(SF61Ø66C)
     SIDELETE
          DESTNAME(SF67583C)
     SIADD
          DESTNAME(SF67583C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67583C)
     SIDELETE
          DESTNAME(SF61128C)
     SIADD
          DESTNAME(SF61128C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF61128C)
     SIDELETE
          DESTNAME(SF67757C)
     SIADD
          DESTNAME(SF67757C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67757C)
     SIDELETE
          DESTNAME(SF67787C)
     SIADD
          DESTNAME(SF67787C)
          TPNAME(RECV)
          MODENAME(LU62C)
          PARTNER_LU(SF67787C)
/*
/*********************************************************************
/*
/* Sample side information for connection FROM remote queue managers
/* to MVS/ESA:
/*
/*********************************************************************
/*   TPNAME   Any value, but must be the same as specified
/*            on the remote queue manager in:
/*             -  the corresponding side information
/*                on MVS/ESA without CICS, OS/4ØØ, or AIX.
/*             -  the channel definitions
/*                on MVS/ESA using CICS or OS/2.
/*   PARTNER_LU
/*            The name of the LU used by the local queue manager.
/*            This LU name must also be specified
/*            on the remote queue manager in:
/*             -  PARTNER_LU in the corresponding side information
/*                on MVS/ESA without CICS, OS/4ØØ, or AIX.
/*             -  the CICS connection definition (that is specifed as
```

```
/*                the connection name in the channel definitions)
/*                on MVS/ESA using CICS
/*             -  the CONNAME attribute in the channel definitions
/*                on OS/2.
/*
    SIDELETE
         DESTNAME(MVSMQLU1)
    SIADD
         DESTNAME(MVSMQLU1)
         TPNAME(MQSERIES)
         MODENAME(#INTER)
         PARTNER_LU(MVSMQLU1)
/*
/*****************************************************************
/*
```

Saida Davies
IBM (UK)

MQSeries for OS/390: extracting the queue name under CICS

How many times have you looked at CICS tasks and wondered which MQSeries queue is being processed? The person who wrote the program may not be available to answer that question. The CICS transaction may be looping, trying to process the same message, or perhaps you are trying to shut down MQSeries cleanly and need to know what is still active.

One way would be to list all queues that are currently open, and, if triggering is being used, check its **PROCESS** definition. This definition would typically state that a CICS transaction should be started and would list its associated CICS transaction id. In fact, to make things easier, it is a good idea to make the name of the **PROCESS** the same as the CICS transaction id.

The queue definition would also show the initiation queue being monitored by the CICS trigger monitor **CKTI**. Again, if you name this initiation queue so that it is similar to the CICS system id, then you would immediately know where the messages are being processed. If the queue does not have a trigger, but it is open, then you have to rely on your system documentation. In my experience, this is usually

missing, or only partly there. Although the supplied **CKQC** transaction shows details of running transactions interfacing with MQSeries, it does not show the queue name (except for **CKTI** itself). Detailed below is a fairly short Assembler/CICS program that will extract the current queue name being used by a CICS transaction. It has been tested under various releases of CICS (4.1 , TS 1.2, and TS 1.3), as well as MQSeries for OS/390 V2.1, and the 'next' release, now known as V5.2.

The output consists of CICS transaction id, transaction number, and the queue, all of which is written to the CICS joblog using a write-to-operator function. You can, of course, change it to send to a BMS map or some other output device. As there are no official interfaces it obtains the information by chasing control blocks, and as such, may need to be changed in the future. Even if you decide not to use it, it can still be valuable for debugging (via CICS transactions CEDF or CEDX) or dump solving. Figure 1 illustrates the control block structure and linkage.

PROGRAM SOURCE

```
**
* FUNCTION : Obtain MQ queue name from a CICS task
* Warning  : Uses Assembler to chase control blocks, and may need
*          : to be altered for another release.
*
* Output   : Output is sent to the joblog in the format
*
*              XXXXTTTTTTTQQQQQQQQQQQQQQQQQQQQ
*
*                  where XXXX is the CICS Transid
*                  TTTTTTT    is the CICS Task number
*                  QQQQQQQ    is the queue name
*
*            You can obviously amend the program so that it is
*            presented in your favourite display format, be it BMS
*            or a Web browser, or whatever.
*
* Tested   : MQ for OS/390 V2.1 and later.
*            CICS V4.1 , TS 1.2, and TS 1.3.
*
* Notes    : Assemble and link as an Assembler/CICS Program
*          :   with AMODE(31) RMODE(24).
*          : Invoke either via CECI LINK PROGRAM('pgmname') or set
```

```
*              up a CICS transaction pointing to the program.
*
R3       EQU    3                        CICS BASE REGISTER
R4       EQU    4                        Work Register
R5       EQU    5                        Work Register
R6       EQU    6                        WORK
R7       EQU    7                        WORK
R8       EQU    8                        WORK
R9       EQU    9                        WORK
R1Ø      EQU    1Ø                       WORK
R11      EQU    11                       CICS EIB BLOCK
R12      EQU    12
R13      EQU    13                       CICS DYNAMIC STORAGE
         EJECT
**********
*    D Y N A M I C  S T O R A G E  A R E A
**********
DFHEISTG DSECT
WSTG     DS   CL4                       Eye catcher 'WSTG'
WLEN     DS   F                         Length of exit work area
@GA      DS   F                         Address of exit work area
@CLOC    DS   F                         Address of CLOC
@CTHD    DS   F                         Address of CTHD
@QRPL    DS   F                         Address of QRPL
@CLOT    DS   F                         Address of CLOT
@MQMD    DS   F                         Address of MQMD
TRANSID  DS   CL4                       CICS Transaction id
TASKID   DS   CL6                       CICS Taskid
QUEUE    DS   CL48                      MQSeries queue name
         DS   ØD                        Used to align packed field
PACKFLD  DS   PL8                       Used to convert taskid
*
         DROP   R13                       DROP R13
         USING  DFHEISTG,R13              R13 -> Dynamic Storage
*-----------------------------------------*
EXMQUEUE CSECT
*
         MVC   WSTG,=C'WSTG'              Start of W-Storage
         EXEC  CICS HANDLE ABEND LABEL(RET)
         EXEC  CICS HANDLE CONDITION ERROR(RET)
*
         EXEC  CICS EXTRACT EXIT PROGRAM('CSQCTRUE')
               ENTRYNAME('MQM') GASET(R4) GALENGTH(WLEN) NOHANDLE
*
         ST    R4,@GA                     Store for Reference
*
         LA    R4,X'1C'(R4)               Find CLOC address
         L     R4,Ø(R4)                   Go to it
         ST    R4,@CLOC                   Store for Reference
*
         LA    R4,X'EC'(R4)               Find first CTHD address
```

```
*
CTHDLOOP EQU   *
         L     R4,Ø(R4)                    Go to it
         LTR   R4,R4                       Test Reg R4
         BZ    RET                         If Zero, no other CTHDs
         ST    R4,@CTHD                    Store for Reference
*
         LA    R4,X'1C'(R4)                Find CLOT address
         L     R4,Ø(R4)                    Go to it
         ST    R4,@CLOT                    Store for reference
*
         MVC   TRANSID(4),X'18'(R4)        Pick up CICS Transid
         ZAP   PACKFLD,X'1C'(4,R4)          and CICS Taskid
         OI    PACKFLD+7,X'ØF'
         UNPK  TASKID,PACKFLD+3(5)         Make Taskid presentable
*
         L     R4,@CTHD                    Go back to CTHD
         LA    R4,X'4C'(R4)                Find QRPL address
         L     R4,Ø(R4)                    Go to it
         ST    R4,@QRPL                    Store for reference
*
         MVC   QUEUE(48),=c'Unknown Queue Name'
         LA    R4,X'8C'(R4)                Find MQMD address
         L     R4,Ø(R4)                    Go to it
         LTR   R4,R4                       Test Reg R4
         BZ    SHOW                        If Zero, no MQMD
         ST    R4,@MQMD                    Store for reference
*
         LA    R5,X'15Ø'                   Set R5=x'15Ø' to find 'OD'
*                                         (Object Descriptor)
LITOD1   EQU   *
         SR    R4,R5                       Go back to find 'OD' literal
         CLC   Ø(4,R4),=C'OD  '            Check we're there
         BE    LITOD2                      Yes, get QName
*
         L     R4,@MQMD                    No, go back to start of MQMD
         LA    R5,X'A8'                    Try x'a8'
         SR    R4,R5                       Go back to find 'OD' literal
         CLC   Ø(4,R4),=C'OD  '            Check we're there
         BNE   SHOW                        No - show unknown name
*
LITOD2   LA    R4,X'C'(R4)                 Go forward X'c'
         MVC   QUEUE(48),Ø(R4)             Pick up Queue Name
*
SHOW     EXEC  CICS WRITE OPERATOR TEXT(TRANSID) TEXTLENGTH(58)
*
         L     R4,@CTHD                    address current CTHD
         LA    R4,X'14'(R4)                Go forward x'14' to next CTHD
         B     CTHDLOOP                    Branch to loop
*
RET      EXEC  CICS RETURN
         END
```

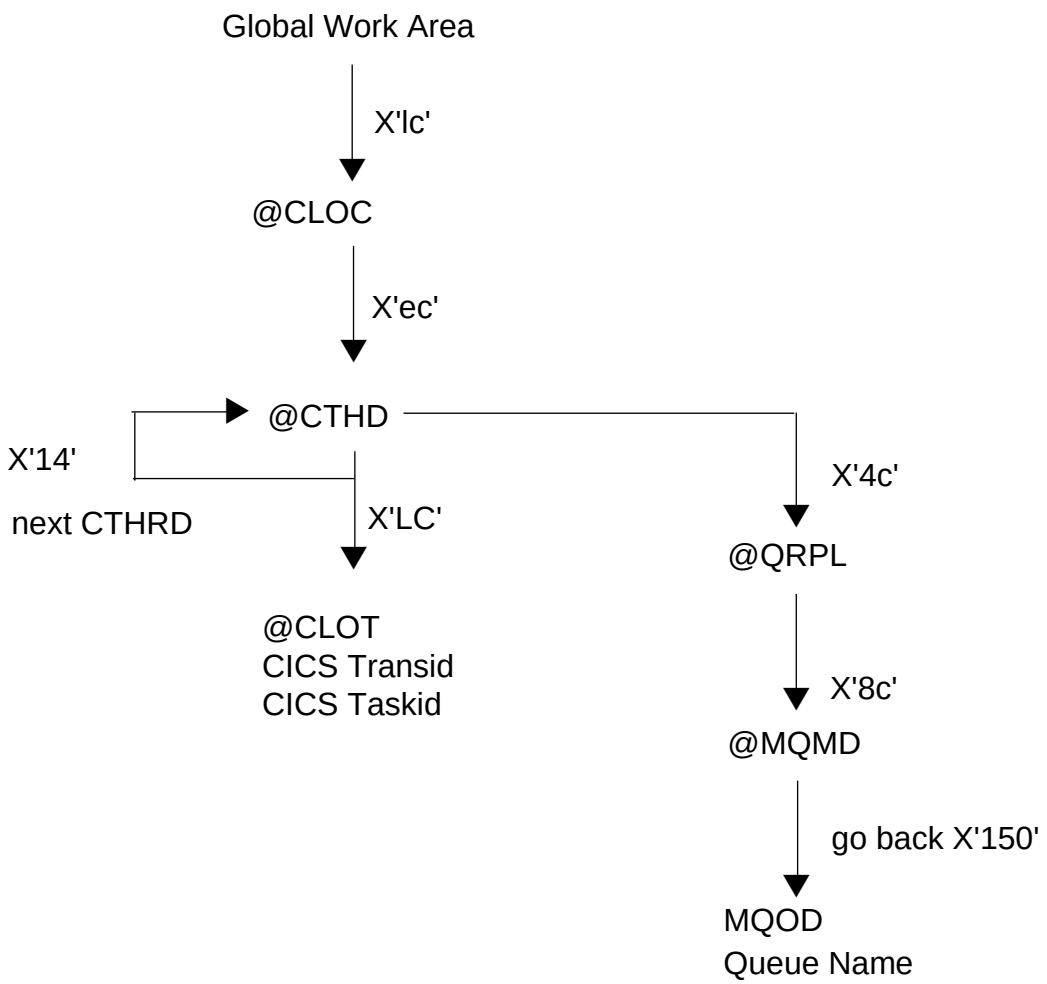

Figure 1: Control block structure and linkage

Please note that, since writing this program, IBM announced the subsequent release of MQSeries for OS/390 V5.2 (October 2000). The new release will provide the above functionality in greater detail, and covering all types of processes, not just CICS. Until the new release becomes available, however, you could use the above program.

Ruud van Zundert
Independent Consultant (UK)

MQSeries and wireless applications

This article explores the wireless world with a specific focus on MQSeries and WebSphere. It looks at the devices available today in the wireless area and the general attributes of each; examines the applications for wireless; discusses some general market information; and provides an architecture overview of four potential wireless techniques.

WHAT IS WIRELESS?

Personal computers

The first type of wireless device to be considered is the notebook computer. While many people today use notebooks, the majority of notebooks are not used for wireless applications. That is, they are more often used for disconnected operation and/or are regularly connected to a network. With the increased availability and bandwidth of wireless modems, these devices will see increased wireless usage. The devices themselves are decreasing in size but increasing in power. Even sub-notebooks are typically configured with Windows 2000.

Palm OS devices

The next group of devices to consider is based on the popular Palm Operating System. These devices, typically referred to as PDAs, are primarily used as personal organizers but are seeing increased usage as data collection devices, replacing paper and pencil. If used in an industrial environment, the device needs to be able to withstand adverse conditions. That is, they will be dropped and jarred on a regular basis – a market need filled by the Symbol devices.

Pocket PC devices

The third group of wireless devices is based on Pocket PC. Pocket PC is the follow-on to Windows CE, which was not overly popular. However, this version has seen very good acceptance and contains features that attract attention, such as advanced audio and video support. These devices will be used in situations similar to those of the

Palm OS devices, but more specifically, when higher-end applications are required. Examples of Pocket PCs include the HP Jornada models 690 and 545, Casio E125, and Compaq iPAQ 3600.

Telephones

The final category of devices discussed is the most prolific and the most diverse. Telephones range in capabilities from basic to advanced. Phones with Web access are the current hot topic. However, there are several implementations of this including:

- iMode, which is heavily used in Japan.
- HDML produced by phone.com. It is the predecessor to WML and is widely used.
- WML is the most commonly-used in Europe and is being implemented throughout the world.
- Java-based phones, which are now becoming available.

Smart phones that include many of the features of a PDA are also based on several operating systems including Palm, GEOS, and EPOC.

UNDERSTANDING YOUR APPLICATION

Before you can begin to develop your application it is most important to know your target customer. Here we discuss three separate potential customer scenarios:

- Manufacturing assembly line or shipping department
- Mobile worker
- Consumer market.

Which of these you are targeting will greatly influence the type of device you will support and the underlying architecture you will choose.

Wireless networks

The first group to consider could be a manufacturing assembly line or

a shipping department. While PCs and PDAs have been used in these environments for some time, they were mainly based on wired networks. Recent developments in the wireless network area will increase the capabilities in this environment.

Bluetooth is a result of joint development that includes 3Com, Ericsson, Intel, IBM, Lucent, Microsoft, Motorola, Nokia, and Toshiba, which provides worldwide wireless connectivity. IEEE 802.11 is another standard specifically for wireless networks. While both of these are slower than wired networks, the opportunity to replace permanently wired devices or eliminate periodic synchronization makes these technologies highly attractive.

It can also be expected that traditionally wired computer users will exploit this technology. For example, an office worker could take a laptop from desk to desk or attend a meeting while maintaining continuous network access.

Mobile workforce

The next potential target is the mobile worker. For example, this could be a salesperson or a claims agent. While these workers typically have notebooks or specialized PDAs today, they primarily use these as collection devices. In many cases, the information collected remotely must be manually re-entered into the corporate systems.

The usage of high-speed wireless modems will allow these users to have immediate access to target systems, greatly improving both their efficiency and customer service. As a consequence, the mobile workforce environment is growing.

By most estimates, mobile workers will outnumber network-attached workers within the next few years. It is worth noting that the typical user will have a notebook PC, a telephone, and most likely a PDA.

Consumer growth

And as a final example, you have to consider the consumer market. While the previous two examples represented millions of users, this group will represent billions. While the size of the market is large, its diversity and complexity are significant factors to consider. This

market will also incorporate some PDAs, Internet-capable phones, and smart phones.

The devices used in this market will have restricted capabilities in that they have small screens, limited input, and slow transmission rates. Most successful applications will target a specific segment with specific deliverables. Ease of use, including voice interaction, is a requirement.

Characteristics

Having identified three potential customers, there are three general characteristics to apply to these customers.

- *Connectivity*. This addresses how a user is connected to a network. The three basic categories here are:
  - always connected
  - periodically connected
  - mostly disconnected.
- *Control*. How much control do you have over the user or their environment? These range from a high degree of control, to a low degree, to no control.
- *Competence*. How competent is the end user in relation to the application being deployed? These users are either highly skilled or have limited skills.

All applications deployed will have some combination of these three characteristics. Applications where the users are highly connected, controlled, and skilled are the easiest to deploy. Applications where the users are typically disconnected, over which you have no control, and which have limited skills will be the most difficult.

Deployment issues

Another aspect to consider is the deployment itself. Here are six significant issues to be considered.

- *Device management*. How much device management is required to deploy the application? This requires some level of control of

the devices. Applications that require no device management are more portable.

- *Security*. There is no doubt today that security is a big issue. Any application deployed must take this into consideration.
- *Scalability*. Given the potential large number of devices, the scalability of an application is important.
- *Infrastructure*. While the end user experience is perhaps the most important, the infrastructure required must be considered.
- *Network reliability*. Depending on the network used, the application will have to consider dropped and intermittent connections.
- *Change*. There is no doubt that this environment will continue to change. An application must be adaptable to changes such as new devices or facilities.

Within each of the following sections, the technique will be evaluated against these issues and awarded marks on a scale of 1 to 5 where 1 indicates a low score and 5 is high.

ARCHITECTURE TECHNIQUES

In this section, we will investigate four potential techniques for implementing applications in a wireless environment. These will include:

- Roll your own solutions.
- MQSeries Everyplace.
- Wireless Access Protocol (WAP).
- WebSphere Transcoding Publisher.

In each section, we will cover the target devices, their characteristics, and how they address the issues identified.

ROLL-YOUR-OWN APPLICATION

As the name implies, ‘roll-your-own’ are locally written or acquired

applications. While the concept of using roll-your-own applications has been common for notebook computers, extending the concept to other wireless devices requires additional consideration for the methods and techniques used.

Figure 1 shows the simplified architecture when developing a wireless-based application. First, you will need to write code that is loaded on the device and typically on a host server. This code provides the tie-in with the rest of the application environment.

Device support

Roll-your-own solutions are applicable to almost all environments. The only exception is devices, such as most phones, which do not allow code to be downloaded. While Java-based phones are becoming

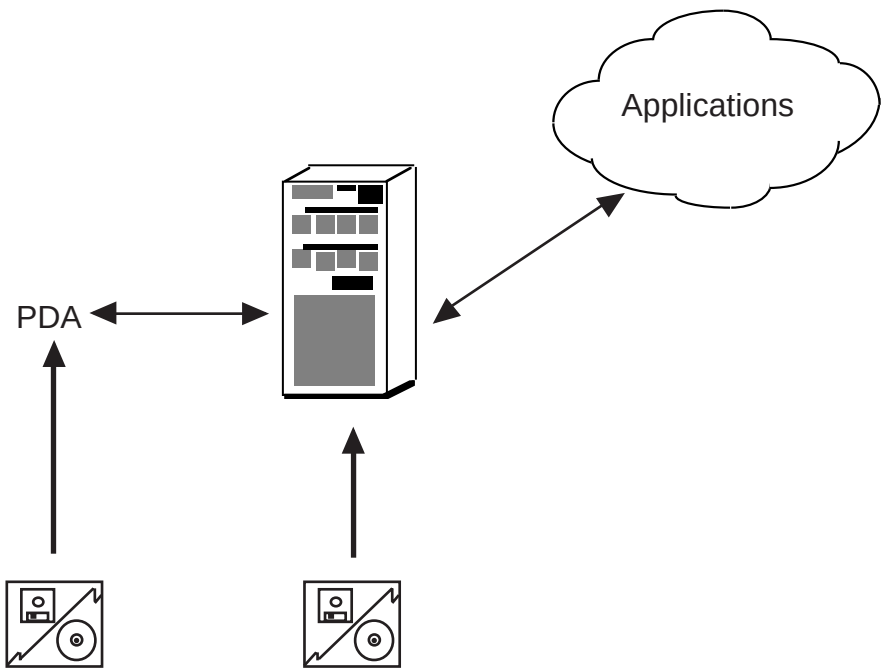

Figure 1: Roll-your-own architecture

available it is still unlikely you would want to write applications that require loading on phones.

Application coding

The code on the device could be written in a product-specific language including C and Java, where supported. The code will interact with the operating system to perform required functions such as data access.

Communication with the serving application is typically done with TCP/IP for interactive applications but may store data locally. Since you have control over both ends of the application, data can be optimized, encrypted, or tailored as required.

The code on the device is similar to other IP-based applications. That is, it simply receives data from the device and processes it. For example, after receiving the data, the application could use the MQSeries API to put and get data from MQSeries queues before responding. Optionally, the host application could be built to take advantage of a synchronization server. Also, it is likely that if you choose the roll-your-own path, you are not actually doing the rolling. A number of software companies specialize in this type of development.

Characteristics

Connectivity

Using this type of application is best when the device is always connected or periodically connected. It may be applicable in disconnected environments but this complicates the application. For example, information could be stored in a database when the device is off-line and then processed in bulk when the device is connected. Or, as another example, if the user needs access to information such as customer data, this would have had to be downloaded before the device was disconnected.

Control

As the device requires code to be installed on the machine, it requires that you have control over the device. For simple applications such as information look-up, you may be able to get by with limited control.

In other cases, where control is required but you don't have it, you can provide a level of control by supplying the device itself.

Competence

This type of application will typically be deployed to end-users that are highly skilled, since they may be required to install and configure the application. They also should be expected to know what your application is trying to perform, because, typically, extended help is not provided**.**

Addressing the issues

- Device management: <**1**..2..3..4..5>

  Device management is a major concern in this environment. Code has to be installed locally that ultimately must be managed (for example when the application changes). Because of the complexity, there are a number of software packages that address device management.

- Security: <1..**2**..3..4..5>

  No specific security structure is provided. User authorization must be performed by the application. However, if control is tight, this could be configured into the machine. Also, if used only on an intranet versus the Internet, some amount of reliability can be assumed, but if not additional security is required.

- Scalability: <1..2..**3**..4..5>

  This solution is moderately scalable. However, application design plays a significant role in determining ultimate scalability.

- Infrastructure: <1..**2**..3..4..5>

  The infrastructure required will include host or synchronization servers. Network configuration is required to connect the devices to the proper hosts. Hosts need to be connected into the native application environment.

- Network reliability: <**1**..2..3..4..5>

  In order to operate in an unreliable environment, your application

must be written to tolerate outages. For example, the application has to decide what to do if the connection is lost in the middle of a transmission or during a multiple-step process.

- Change: <1..**2**..3..4..5>

  Because the application is device-specific using device-provided facilities, it is difficult to adjust for new facilities or devices. Application design can become complex as the number of supported devices increase.

MQSERIES EVERYPLACE

MQSeries Everyplace is a version of MQSeries designed specifically for small footprint devices. It is built on the same philosophy as other MQSeries products but is a unique code base with a unique API. In addition to wireless devices discussed here, it is also seen as being implemented on other pervasive devices such as sensors and appliances. Once and once only delivery as found on other MQSeries platforms is provided.

Figure 2 shows the simplified architecture when deploying an application using MQSeries Everyplace. There are three components to the application:

- The device on which MQSeries Everyplace and the application are loaded.
- A gateway device that facilitates connecting the many devices into a single point, can hold messages in transit, and is the access point to the MQSeries network.
- The Server that represents the connection with the rest of the MQSeries environment.

A unique aspect of MQSeries gateway is native support for device-to-device connectivity.

Device support

The list of supported devices include:

- Palm OS

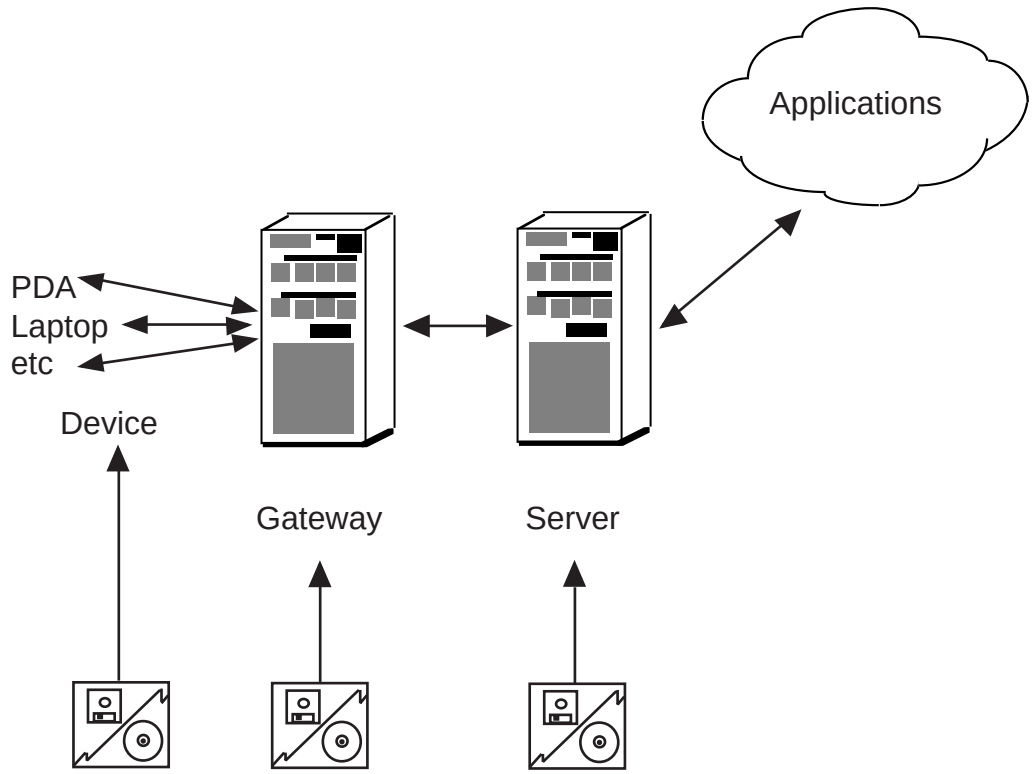

Figure 2: MQSeries Everyplace architecture

- EPOC
- Windows
- Pocket PC, WinCE, 95, 98, NT, and 2000
- Any Sun-certified Java VM 1.1
- IBM Visual Age for Java Micro Edition.

Gateway support

With the latest release, MQSeries Everyplace for Multiplatforms

v1.1, announced in October, the platforms supported for the Gateway include:

- AIX
- Sun Solaris
- Windows NT
- Windows 2000.

Server support

The Server can ultimately be any of the 35 platforms supported by MQSeries.

Application coding

The application developed will be in Java except on the Palm OS devices that currently only support a C interface. The application will use an object-oriented Java-based API into MQSeries Everyplace that supports similar but different facilities from traditional MQSeries. For example, synchronous as well as asynchronous applications are supported. Messages can be queued on the local device or be remote on another device or the gateway. With asynchronous messaging, control will be returned to the application as soon as the device processes the request.

With synchronous messaging, control is not returned to the application until the message is delivered into the target queue. Palm OS devices cannot hold local messages and accordingly,only support synchronous messaging.

One of the major values of MQSeries Everyplace is that it provides the communication component. This allows it to provide optimized data streams as well as encryption.

However, one of the implications of applications developed using MQSeries Everyplace is that the device may not transmit or have access to, all of the data required. This is handled by code in the gateway to augment the message with the data required. In fact, it is possible that the messages sent into the MQSeries environment might still need additional augmentation or special processing.

Connectivity

Because of its design, MQSeries Everyplace is suited for all connectivity types. However, for mostly disconnected devices, applications would most likely use local messaging, eliminating the Palm OS devices. Synchronous messages between devices would be impractical if the devices were primarily disconnected.

Control

The requirement to install MQSeries and application code plus configuration would require that these devices be highly controlled.

Competence

These applications would most likely be targeted towards skilled individuals.

Addressing the issues

- Device management: <**1**..2..3..4..5>

  In addition to the requirements of a roll-your-own solution, you would be required to deploy MQSeries Everyplace. For pervasive devices, this could be embedded in the device, but for customer applications, management of the device will be required.

- Security: <1..2..3..**4**..5>

  One of the key aspects of MQSeries Everyplace is the security features. Encryption is the strongest component, making MQSeries Everyplace a good choice for Internet-based applications.

  There is a lack of standards for certificates for low bandwidth devices, requiring some consideration of how to secure these transmissions. Once into the native MQSeries environment, deriving an equivalent user identity may be difficult. Security can be configured in such a way that decrypted queued messages could not be read locally, preventing hacking into the device if it was lost or stolen.

- Scalability: <1..2..**3**..4..5>

  The design of MQSeries Everyplace is scalable. IBM recommends

100 devices per gateway for moderate workloads and up to 500 for lighter workloads.

- Infrastructure: <1..**2**..3..4..5>

  One of the current weak areas of MQSeries Everyplace is the complexity of the infrastructure. Configuration is required at the device and at the gateway. Existing MQSeries experts may actually be confused by the new concepts.

- Network reliability: <1..2..3..**4**..5>

  Once and once only assured delivery of messages is the primary attraction of MQSeries Everyplace. Messages will either be transmitted completely or not at all. The application could inform the user if connectivity was lost before a given action was completed. Depending on the design, the application will still be responsible for failure within multiple interactions.

- Change: <1..2..**3**..4..5>

  MQSeries Everyplace provides a layer of insulation between the application and the communication protocols. The use of a Java-based API allows the application to be insulated from platform dependencies and allows the application be ported to any device that supports a compatible JVM.

An alternative usage

One interesting aspect of MQSeries Everyplace is actually not for wireless usage at all. Consider that MQSeries Everyplace supports device-to-device communication and that supported devices include Windows NT and Windows 2000. In conjunction with the compression and encryption capabilities provided, this makes MQSeries Everyplace an alternative for any high security application.

Consider two business partners that want to share information over the Internet. Native MQSeries does not provide encryption, thus requiring the applications to handle this themselves. However, by using MQSeries Everyplace, encryption is provided. The only negative is that, because MQSeries Everyplace API is similar but not the same as MQSeries, these applications will have to be specifically written for this purpose.

Wireless Access Protocol, most commonly known as WAP, is everywhere in the press these days. All of the major phone manufactures in Europe (and the US) are delivering WAP-compatible phones. WAP applications are built using WML, an XML HTML derivative. Any device with a WML-based browser can be used to access WAP applications.

While a relatively new standard, WAP has also been widely accepted by application development products and a number of WAP-based services are already on the market. Unfortunately, one of the disadvantages of WAP is the small display size and the difficult interaction required. Many applications written to date have not taken this into account and WAP has received some negative reviews.

Figure 3 displays the basic architecture of a WAP application. The devices connect to a WAP gateway (also referred to as a WAP server or proxy).

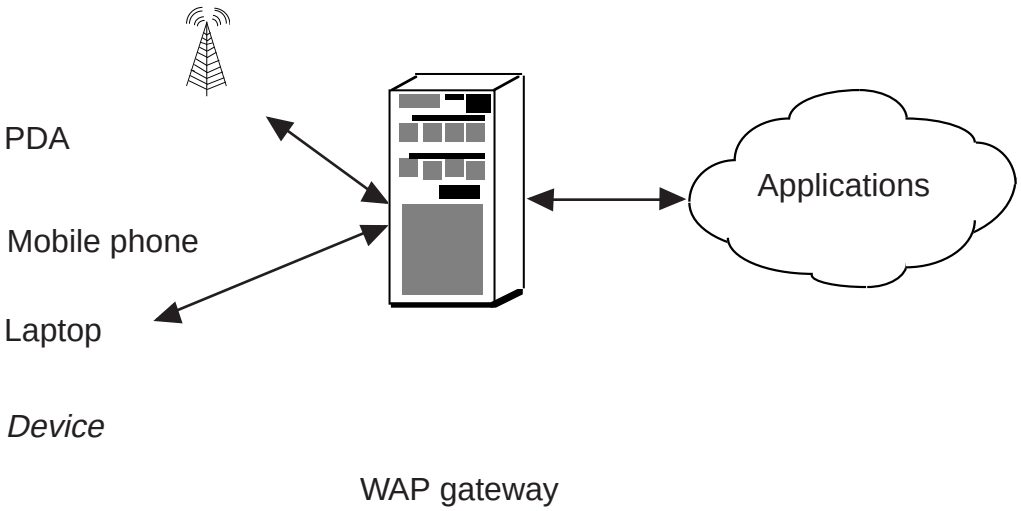

Figure 3: WAP architecture

The gateway converts the communication from a binary WAP format into WML, which is communicated to the application over HTTP. Because the devices are unable to handle state information (such as cookies), the gateway may also keep some state information locally and may cache some information to improve performance.

Communication carriers as well as a number of vendors are offering WAP Servers. The WAP server could be installed at the carrier site or within a corporate environment.

Application coding

One of the major advantages of WAP over the previous two application types is that no coding is required on the device itself. The only requirement is that the device supports a WML-capable browser. Additionally, the gateway does not require any code. And in theory, any application could be used to talk to a WAP device.

Many of the application generators that use HTML have been modified to be WML-capable. Many HTML/WML converters are also available. This provides the benefit of the same back-end application being used for both HTML and WML interfaces. Unfortunately, it is unlikely that an application written for a network-attached device with a large display will work well without modification on a WAP device.

Connectivity

WAP works very well with all types of device, although devices that are always connected may be a better fit for other techniques.

Control

The lack of any local code or data on the device allows WAP to be delivered on any device with no requirement for control.

Competence

WAP applications should be designed for quick and easy access. While acceptable for skilled workers, the small display size is not suitable for many applications where the end user requires a lot of interaction.

Addressing the issues

- Device management: <1..2..3..**4**..5>

  Basically, no device management is required, with the exception of physical management to distribute and manage the devices.

- Security: <1..2..3..**4**..5>

  WAP provides a solid security model. Data transmitted over the carrier network is encrypted and data between the gateway and the application server is also encrypted.

  However, a current security exposure exists at the WAP gateway. Since the gateways and encryption schemes are not the same on both sides, the gateway must decrypt/encrypt the message. It is possible for someone with access to the gateway to see the decrypted data and potentially alter it. If the gateway is hosted by a third party, this may be an unacceptable exposure.

  There is also no current support for digital certificates or smart cards from the devices themselves.

- Scalability: <1..2..3..4..**5**>

  By design, WAP solutions are intended to be scalable.

- Infrastructure: <1..2..3..**4**..5>

  The fundamental WAP architecture is basic, requiring only the addition of WAP gateways to the existing environment. Typically, the WAP gateways are not process-intensive, so each server should be able to handle a large number of devices. The gateways may require some set-up, which can become a major task. This includes setting up users, devices, and accessible URLs.

- Network reliability: <1..2..**3**..4..5>

  While WAP itself will tolerate unreliable networks, there is nothing inherent in the protocol to provide ensured message delivery as provided in MQSeries Everyplace. That is, if the customer presses OK and the line disconnects, was the OK sent or not? Application synchronization and multiple interaction processing are required.

- Change: <1..2..3..**4**..5>

  Coding using WAP requires the least device-dependence at the expense of reduced functionality. Unfortunately, not all devices implement WAP in exactly the same way, so care must be taken to ensure that features work as expected on all target devices. Using WAP enables new devices or facilities to be incorporated easily. The major exposure is the replacement of WAP by another technique.

WEBSPHERE TRANSCODING PUBLISHER

Hopefully, one of the apparent disadvantages of all the techniques discussed to this point has been that the applications, techniques, and devices are all intertwined. The devices provide a wide range of capabilities, from no graphics to high-resolution graphics, from low bandwidth to high bandwidth, from very limited input to versatile input.

Ideally, an application should be able to use any type of device, taking advantage of the device characteristics. This is the goal of WebSphere Transcoding Publisher.

The architecture displayed in Figure 4 is similar to the preceding techniques. A server is positioned between the application and the device. WebSphere Transcoding Publisher uses device and application profiles to customize the interaction between them.

Other architecture configurations include:

- The use of a cacheing server to hold commonly-used transcoding, which can significantly improve performance.
- Support for WebSphere application server without the proxy server if the application runs in this environment.
- Direct access using Enterprise JavaBeans (EJB) without the proxy server for Java applications.

Application development

No device-level code is required. However, it is possible to implement

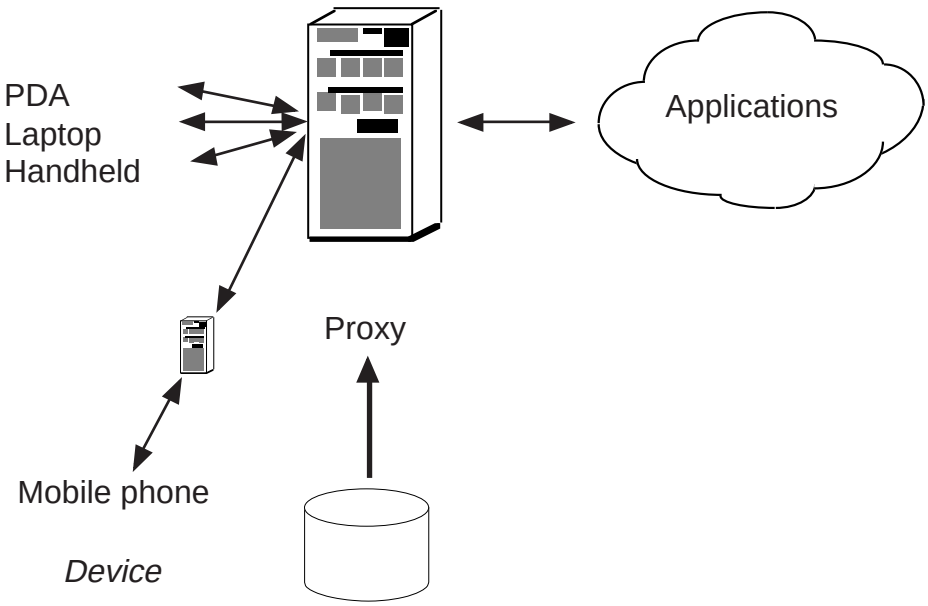

Figure 4: Transcoding architecture

device code that is connected into the application via WebSphere Transcoding Publisher. Using MQSeries Everyplace to perform assured messaging is an example of where this could be used.

At the proxy server, profile and style sheets are created to define application dependencies.

From the application perspective, if the goal of transcoding is met, the application is unaware of the device being used.

Using the administration functions, profiles are created to convert (transcode) the interaction between the application and the device. Profiles are based on the user, the device, network (port), or Web address. Profiles consist of a sequence of steps that modify the data stream. For example, images, unsupported tags, and scripts could be removed (simplification). Text could be edited to remove unnecessary portions (clipping). Images could also be reduced to lower resolution

and trimmed to a specific portion (scaling). For WML devices that support a small total message size, the data can be broken into multiple segments (fragmentation).

Device support

The devices supported by WebSphere Transcoding Publisher include:

- Desktops/notebooks
- Palm
- Pocket PC
- WAP devices
- iMode devices
- HDML devices.

Support for Palm and Pocket PC is restricted to browser access only.

Server support

For other than initial testing, the WebSphere Transcoding Publisher Server needs to be installed on a dedicated machine running one of the following:

- Windows NT or 2000
- AIX
- Sun Solaris
- Red Hat Linux.

Additionally, it requires JDK 1.2.2 to be installed. Note that this level of Java is not universally supported by all IBM products.

Connectivity

WebSphere Transcoding Publisher is the most tolerant to connectivity aspects. In fact, it can exploit these characteristics much more than any of the other techniques.

Control

While WebSphere Transcoding Publisher is flexible from a control perspective, the creation of style sheets and profiles becomes increasingly difficult if some control is not enforced.

Competence

The flexibility of WebSphere Transcoding Publisher allows device applications to be tailored to the device and the potential user.

Addressing the issues

- Device management: <1..2..3..**4**..5>

  Limited device management is required. However, creation of profiles is somewhat dependent on the devices in use.

- Security: <1..2..**3**..4..5>

  WebSphere Transcoding Publisher provides a number of levels of security in addition to that provided if WAP is used to communicate to the device. However, only when running the proxy server as a WebSphere application is encryption supported.

  This means that the normal usage of WebSphere Transcoding Publisher requires that the WAP server and the proxy are installed within a controlled firewall.

- Scalability: <1..2..3..**4**..5>

  The application should be scalable using this technique. WebSphere Transcoding Publisher servers are required.

- Infrastructure: <1..2..**3**..4..5>

  Profile creation and management will not be a trivial task. Device and application changes will be an ongoing process. Image and text clipping is very dependent on the application.

- Network reliability: <1..**2**..3..4..5>

  WebSphere Transcoding Publisher does not provide any specific support to address unreliable networks.

- Change: <1..2..3..4..**5**>

  In theory, WebSphere Transcoding Publisher is developed to embrace change. New devices or facilities could be integrated into the application without changes to it. Support for iMode devices is an example of this.

WEBSPHERE EVERYPLACE SUITE

WebSphere Everyplace is a recently announced suite of products, of which the flagship products include WebSphere Transcoding Publisher, MQSeries Everyplace, a wireless server, and a WAP Server. As such, the suite provides the 'Swiss army knife' for wireless development.

Depending on your application, you can choose or combine the relevant pieces to provide the required level of function.

In addition to the components discussed here, the package also contains Tivoli Internet Subscription Management, a Device Management server, and a synchronization server. The suite comprises a total of 11 CDs.

Optionally, voice support is available. Voice support is a key deliverable to make voice-based applications easy to use.

SUMMARY

In summary, the key to a successful wireless implementation is first understanding your application and the target audience as well as the infrastructure you will use.

Next, match these against the features and facilities available. Then develop and deliver your application, meeting time-to-market and ease of use goals.

However, this is a highly dynamic market and things will change, so be ready to consider what will happen next.

Note that if you are using WebSphere Everyplace Suite you can combine the benefits of all the techniques for most features. However, it's not always going to be additive; for example, using MQSeries

	Roll your own	MQSeries Everyplace	Wireless Access Protocol	WebSphere Transcoding Publisher
Device management	✔	✔	✔✔✔✔	✔✔✔✔
Security	✔✔	✔✔✔✔	✔✔✔✔	✔✔✔
Scalability	✔	✔✔	✔✔✔✔	✔✔✔
Infrastructure	✔✔	✔✔	✔✔✔✔	✔✔✔
Unreliable networks	✔	✔✔✔✔	✔✔✔	✔✔
Change	✔✔	✔✔✔	✔✔✔✔	✔✔✔✔✔

Figure 5: Comparison of the varying techniques

Everyplace with WebSphere Transcoding Publisher provides the added reliability and security but still requires the infrastructure for MQSeries Everyplace.

Using the additional components, such as the Tivoli Internet Subscription Management and Device Management Server may reduce infrastructure and device management requirements depending on your requirements.

Figure 5 provides a comparison of the various techniques discussed.

Richard G Nikula,
BMC Software

IBM MQSeries professional certification

INTRODUCTION

The value of professional qualifications in the computing industry is a contentious issue but there is no doubt that they are here to stay. Five years ago, IBM introduced the Certified MQSeries Engineer qualification. This has since mutated into three other qualifications, and more have subsequently arrived, too.

IBM's motivation for introducing MQSeries certification is debatable. Back in 1995, it was difficult for the company to achieve sales as customers were rightly concerned about the availability of skilled support staff. Also, there was concern that the imminent arrival of Microsoft's MSMQ product would provide serious competition.

Microsoft already had a certification programme, which enabled it to boast about the number of people with skills in its products. IBM's introduction of MQSeries certification helped alleviate this difficulty.

However, Peter Goss of MQSeries Product Marketing has responded to suggestions about the tests being a sales ploy, saying "Your statement is not correct. The tests in the MQSeries Family Certification programme – all eight of them – were introduced to allow individuals to validate the success of the education they have taken and experience they have gained working with the products. And it allows business partners and customers to validate that their employees have gained the education, knowledge, and experience necessary to do their jobs."

There is also debate over who should pay the costs of sitting the tests.

It is in IBM's interests to have lots of qualified people in the market. If employers demand that job candidates are certified, then people have no choice but to take the tests, and often to attend supporting IBM education programmes.

In order for a software vendor or consulting company to receive support from IBM it must become an IBM Business Partner. One of the requirements for a company to be a member of IBM's PartnerWorld programme is that they have a certain number of employees who are

certified.

Peter goes on to say, "Tests, therefore, have real value to an individual or to a company, and so it's no surprise that it costs money to take the test".

Another area of concern is that employers could use the fact that an employee has failed the test as justification for dismissal. For this reason, some employees refuse to take the test until they are absolutely sure that they are ready, but this can cause conflict.

Having said all this, the Certification Programme is here to stay, and if you achieve the qualification you are undoubtedly in a better situation than if you do not.

WHAT IS THE IBM PROFESSIONAL CERTIFICATION PROGRAMME?

The IBM Professional Certification Programme is the roadmap provided by IBM such that you can achieve internationally recognized certification by attending education classes and taking examinations that demonstrate your abilities and experience. IBM says that its Professional Certification Programme "Offers a business solution for skilled technical professionals seeking to demonstrate their expertise to the world".

The programme is designed to validate skills objectively, and demonstrate proficiency in the latest IBM technology and solutions. The company says that by giving you and your employer confidence that your skills have been tested, the certification can help you excel at your job, delivering higher levels of service and technical expertise, and thus move you onto a more progressive career track.

For optimum benefit, the certification tests must reflect the critical tasks required for a job, the skill levels of each task, and the frequency with which a task needs to be performed. IBM has comprehensive documented processes, which ensure that the certification tests remain relevant to the work environment of potential certification candidates.

IS IT WORTH THE MONEY?

A typical price for a test is £105, although it is sometimes possible to

get a ‘You pass we pay’ deal depending upon your position (eg business partner). Sometimes, at IBM conferences, it is possible to sit the test for free. The prerequisite IBM MQSeries Courses typically cost around £300 per day, and last between one and five days.

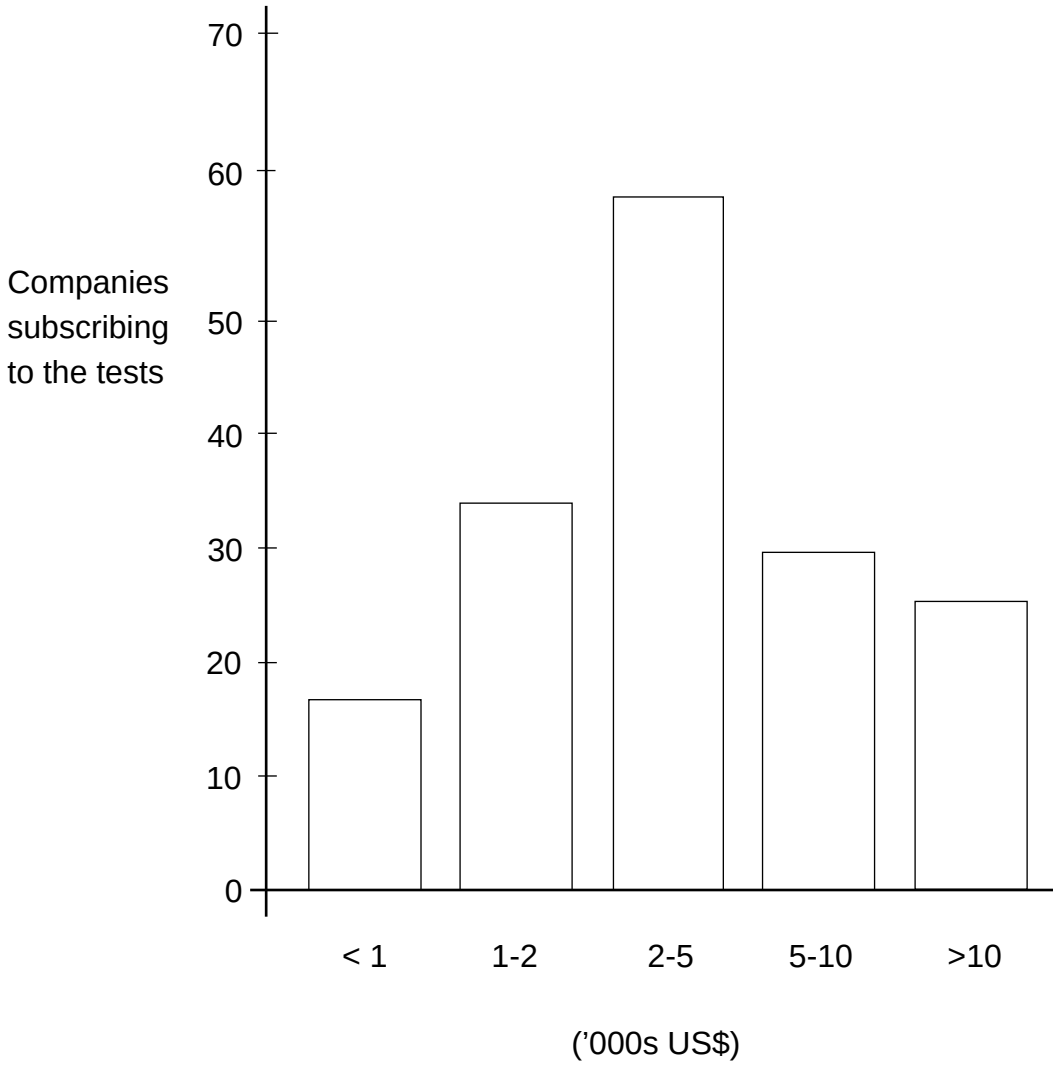

Figure 1: Amount of money invested in certification per individual

(Source: IBM)

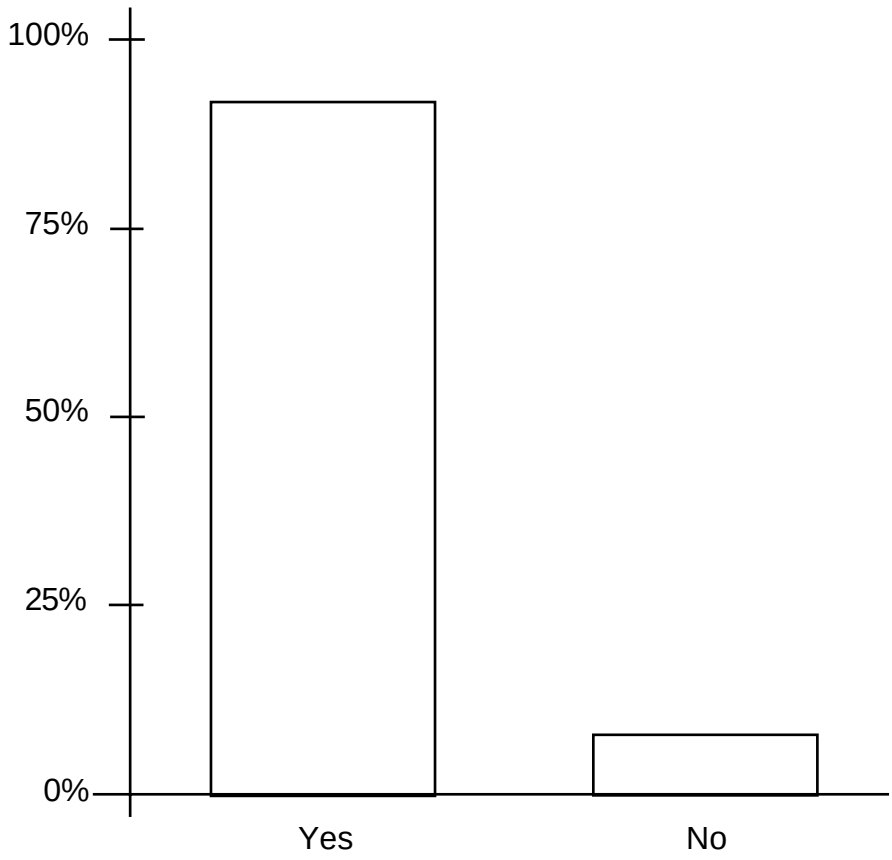

Figure 2: Does the investment yield the anticipated ROI?

(Source IBM)

IBM justifies the cost of taking the test and associated education with a Return on Investment study (Figures 1 and 2). They say that the study indicated positive business results for the companies taking the test, such as improved revenue (profitability), efficiency (productivity), and customer satisfaction (credibility).

THE PATH TO CERTIFICATION

The process by which certification is achieved is shown in Figure 3.

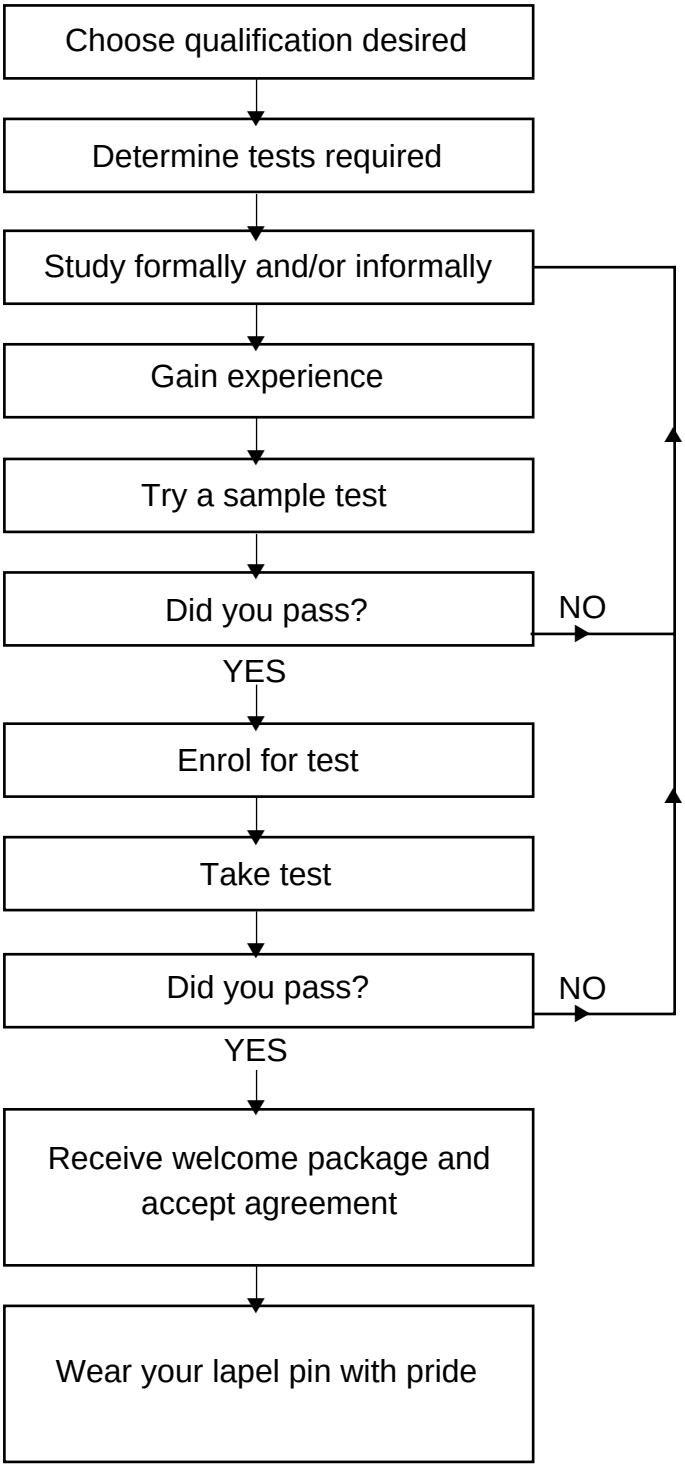

Figure 3: The path to certification

WHAT MQSERIES-RELATED QUALIFICATIONS ARE AVAILABLE?

At Level 1 you can take tests to qualify as an IBM Certified Specialist in MQSeries, MQSeries Integrator, or MQSeries Integrator V2. This title certifies you to perform basic operational services, such as basic planning, configuration, installation, support, management, and maintenance, with limited assistance, or to perform administration of the product, with limited assistance.

At Level 2 you can qualify as an IBM Certified Solutions Expert in the same products and also in MQSeries Workflow. This title demonstrates breadth of basic operational services skills in more than one environment, or demonstrates depth of advanced operational services skills, such as customizing, integrating, migrating, and tuning, in one environment.

Also at Level 2 is the IBM Certified Developer qualification for MQSeries. It demonstrates the capability to plan and design an application requirement and build a prototype.

As an example, the people who should consider applying to be IBM Certified Solutions Experts in MQSeries are people responsible for planning and architecting software solutions and designing applications based on MQSeries.

They should have the knowledge available from attending the MQSeries Technical Introduction course, along with the Application Programming course, the Domino Connections course, and the Advanced System Design and Connections course.

They should also have practical experience of implementing MQ with transaction management and database products, systems management, prototyping IT solutions, basic programming concepts, IT security concepts and skills in implementing systems on multiple computing platforms, plus the ability to gather business requirements and to translate those requirements into IT solutions.

WHICH TESTS ARE REQUIRED?

Each of the qualifications requires the taking of at least one test, with the recommended prerequisite study and experience (see Figure 4).

Certification	Test	Prerequisite qualification
MQ Specialist	MQ Installation and Configuration	
MQ Solutions Expert	MQ System Planning and Design	
MQ Developer	MQ Application Design	
MQSI Specialist	MQSI Installation and Configuration	MQ Specialist
MQSI Solutions Expert	MQSI System Planning and Design	MQ Solutions Expert
MQSI V2 Specialist	MQSI V2 Implementation	MQ Specialist
MQSI V2 Solutions Expert	MQSI V2 System Architecture and Design	Any MQ certification
MQSeries Workflow Solutions Expert	MQSeries Workflow System Planning and Modelling	

Figure 4: Recommended prerequisite study and experience for taking MQ tests

The mapping between the qualifications and test names is not immediately obvious but it does help you to understand what the qualifications signify.

Details of the individual test objectives, describing the topics potentially covered on the tests, can be found at *http://www.ibm.com/education/certify/tests/index.phtml*.

WHAT STUDY IS AVAILABLE?

Each test specifies, as a prerequisite course, the MQSeries Technical Introduction, which is available from IBM Learning Services as a class lesson or as a computer-based training CD-ROM.

They each require at least familiarity with the MQ manuals and publications, and knowledge of basic MQSeries functions and facilities. Each certification also has other specific courses associated with it, the details of which are available on the tests' Web page.

EXPERIENCE

This, of course, is the contentious part. The qualification is designed to show that you have adequate experience to cope with real-life situations. Taking courses designed to get you through the test and missing out the real-life experience is defeating the object.

SAMPLE TESTS

When you think you may be ready to take the test it is worth trying a sample one. This way, you will know whether it's worth spending the time, money, and stress on the real thing. Also, you'll get a feel for the style of questions and may even learn more through the experience. Sample tests are available from the Web site *http://www.ibm.com/education/certify/tests/*.

There is also a similar test at *http://www-4.ibm.com/software/sw-sell/besteam/catalog/qualtest/mqs_qual.htm*, which can be used as preparation, although it is actually for the IBM Business Partner Software Programme.

Example question

Here is an example question. It is a very simple question but the answer is not so easy. See the Web site for the answer.

An MQSeries application has created a queue with the following conditions specified on the DEFINE QLOCAL command:

```
DEFPRTY(0)
MSGDLVSQ(FIFO)
```

```
TRIGMPRI(5)
TRIGTYPE(DEPTH)
TRIGDPTH(10)
TRIGGER
```

When will a trigger message be generated?

A No trigger messages will be generated.

B When the queue contains 5 messages.

C When the queue contains 10 messages.

D When the queue contains 5 priority messages.

E When the queue contains 10 priority 5 messages.

It is common to say that IBM has simply got the answer wrong and/or that there is more than one answer. This is a common criticism of the tests in general but usually shows that the person has not fully understood the question or the reason for the answer.

This is a good way to identify which people deserve the certification, but unfortunately gives the test a bad reputation.

ENROLLING FOR THE TEST

Tests can be scheduled with either an IBM Learning Centre or Prometric Inc (formerly Sylvan Prometric). IBM Learning Services can be contacted on 0845 758 1329 (in the UK).

It may be possible to register as late as the day before the test. You will be required to pay at the time of scheduling and, if time permits, you will be sent a letter of confirmation.

Taking the test itself

You will need to take two forms of identification. At the conclusion of the test you will receive a full score report with section analyses. If you fail you will be able to register for another test, which should have different questions.

THE CONTENT OF THE TESTS

Each of the MQSeries tests has questions that relate to MQSeries

Version 5.1 for distributed platforms and MQSeries for OS/390 Version 2.1. Each test has between sixty and eighty questions and is of multiple-choice, closed book format. Most last ninety minutes but some are seventy-five minutes.

Some people say it is possible to pass and still know very little, or to study, cram, pass the test, and then forget it. Some object to the tests covering areas that are not relevant to their work; however, you do not need to get 100% to pass, so it is not expected that you will know all the areas of the product. It shows that you have a reasonable amount of knowledge.

A common criticism of the tests is that they measure retained knowledge, and people say that, in real life, they would not need to know it, that they would look up the information in a manual or call IBM.

The certification specifications clearly state that they certify that a person is self-sufficient, performing the tasks with limited assistance from peers, product documentation, and vendor support services.

There is only so much that can be achieved by someone who needs to rely on documentation and support. In the crisis situation of a live production problem there may not be the time to look things up or to wait for a call to be returned. An employer will be looking for someone who can cope under this sort of pressure.

Also, it is necessary to have complete understanding and retention of fundamentals to be able to understand the more complex MQSeries and application scenarios. For example, it would not be possible to master MQSI if you had to keep referring back to MQSeries itself. However, the tests also cater for the other point of view by asking where you might expect to look up certain information.

THE WELCOME PACKAGE

Between three and six weeks after passing the test IBM will send you the Certification Agreement and a welcome package, which includes your certificate. After accepting the Certification Agreement you may open the package and use the IBM Professional Certification title and trademark.

The package contains a certificate, a wallet sized certificate, a lapel pin, and details of the certification logo and how you can use it.

You are asked to keep the testing centre informed of any changes in your personal details. They in turn keep IBM informed. Apparently, this means that they can keep you informed of all programme information. However, I have never received anything other than the original pack.

CONCLUSION

MQSeries skills are still scarce at the moment and so people may not feel the need for certification, but as more people learn MQ skills certification may become more useful. It is in the interests of both employers and good employees to find a way to filter out the people who claim abilities that they do not have. The certification system is by no means perfect but it is still a good step towards achieving this.

Sam Garforth
IBM Certified Specialist: MQSeries
IBM Certified Developer: MQSeries
IBM Certified Solutions Expert: MQSeries

Code from *MQ Update* articles

As a free service to subscribers and to remove the need to rekey the scripts, code from individual articles of *MQ Update* can be accessed on our Web site, at

http://www.xephon.com/mqupdate.html

You will need the user-id shown on your address label.

MQ news

Computer Associates has begun shipping Version 9 of its Aion business rules automation engine, enabling non-technical business managers to directly and immediately modify application business rules.

Version 9 provides integration with MQSeries and other development technologies and tools.

There's a new Web interface option on top of the existing client/server implementation. There's also integration with CA's Neugents and Jasmine *ii* software.

Integration with Jasmine *ii* means Aion applications can now tap into databases, applications, objects, and services, adding rules-based intelligence to each. Aion components can also be accessed through Jasmine ii, so that any development tool or client supported by Jasmine can now take advantage of Aion functions.

For further information contact:
Computer Associates, 1 Computer Associates Plz, Hauppauge, NY 11749, USA.
Tel: +1 516 342 5224
Fax: +1 516 342 5329
Web: http://www.cai.com

Computer Associates, Ditton Park, Riding Court Road, Datchet, Slough, Berkshire SL3 9LL, UK.
Tel: +44 (0)1753 577733
Fax: +44 (0)1753 825464

* * *

Information Builders has announced support for IBM's WebSphere Application Server, specifically working with the data and application access of Advanced Edition 3.5 through its new Enterprise Connector for IBM WebSphere Application Server.

The firm brings support to more than 120 relational, non-relational, transactions, and application sources on 35 platforms, including MQSeries and MQSeries Integrator-based messaging applications, popular relational and legacy databases, file systems, CICS and IMS/TM transaction systems, SAP R/3, PeopleSoft, and J D Edwards application environments, and custom-developed applications written in COBOL. RPG, C, and C++.

For further information contact:
Information Builders, 2 Penn Plz, New York, NY 10121, USA.
Tel: +1 212 736 4433
Fax: +1 212 967 6406
Web: http://www.ibi.com

Information Builders, Wembley Point, Harrow Road, Wembley, Middx HA9 6DE, UK.
Tel: +44 20 8982 4700
Fax: +44 20 8903 2191

* * *