Introduction to Accounting SMF data

Lyn Elkins – IBM ATS

elkinsc@us.ibm.com

Agenda

- **What is SMF?**

- **What is MQ SMF?**

- **Overview of MQ accounting SMF**

- **Controlling the generation of the data**

- **Processing the data**

- **What the reports look like**

- **Simple examples of problems found**

What is SMF?

- System Management Facility

- Defined on Wikipedia as
  - **"IBM System Management Facility (SMF)** is a component of IBM's z/OS for mainframe computers, providing a standardised method for writing out records of activity to a file (or data set to use a z/OS term). SMF provides full "instrumentation" of all baseline activities running on that IBM mainframe operating system, including I/O, network activity, software usage, error conditions, processor utilization, etc. "

  - "SMF forms the basis for many monitoring and automation utilities. Each SMF record has a numbered type (e.g. "SMF 120" or "SMF 89"), and installations have great control over how much or how little SMF data to collect. Records written by software other than IBM products generally have a record type of 128 or higher. Some record types have subtypes - for example Type 70 Subtype 1 records are written by RMF to record CPU activity. "
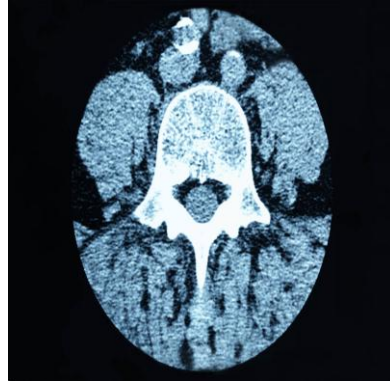
What is MQ SMF?

- MQ generates two SMF types:
  - MQ Statistical data – SMF115
    - QMGR level resource usage and activity
    - Contains information from the resource managers
    - Two Subtypes:
      - Subtype 1 includes the storage manager and log manager information
      - Subtype 2 includes the message manager, data manager, buffer manager, lock manager, DB2 manager, CF manager, topic manager and SMDS manager information
  - MQ Accounting data – SMF116
    - Task level resource usage and activity
    - Contains information from each task that uses the queue manager
    - Three Subtypes:
      - Subtype 0 includes the message manager information
      - Subtype 1 includes thread level identification, thread level accounting, and queue level accounting
      - Subtype 2 includes additional thread level accounting and queue level accounting (if needed)

What is the MQ Accounting SMF used for?

- Checking the detailed health of the queue manager
- Determining the application use of MQ resources
- Detailed performance review
- Detailed problem determination
- Capacity planning

© 2013 IBM Corporation

What is the MQ Accounting SMF used for?

N
O
T
E
S

- Checking the detailed health of the queue manager
  - For example, I know that bufferpool 2 is being heavily used, but I don't know which applications are using that resource.
- Determining the application use of MQ resources
  - Is this application using more CPU than it should because of the coding techniques?
- Detailed performance review
  - Looking at each task and it's use.
- Detailed problem determination
  - Why am I missing my SLAs every Monday at 2:30 pm?
- Capacity planning
  - Even though I am not having any problems currently, where might I need more capacity if In increase workload of existing transactions by 20% during my peak period?
  - I am going to be adding a new application, which impact will that have on my existing resources?

Gathering the data

- Turning on the data collection:
  - ▶ SMFACCT=YES in the CSQ6SYSP macro does not turn on the class 3 trace
  - ▶ +cpf START TRACE(ACCTG) CLASS(3) DEST(SMF)

- Timing the record generation
  - ▶ SMF stats are always collected
  - ▶ Records are generated:
    - At STATIME if specified in the CSQ6SYSP
      - – STATIME=15 will generate the SMF records every 15 minutes
    - At the system SMF interval if not specified or set to 0
      - – STATIME=00
      - – z/OS SMF parameter INTVAL(60) – will generate the SMF records every hour

- Where is the data stored
  - ▶ Traditionally the 'MAN' datasets
    - MPXCAT.SMF.&SYSNAME..MANX
  - ▶ System logger
    - High volume customers have found benefits to this technique

Gathering the data - Notes

N
O
T
E
S

- The SMF116 class 3 data is copious
  - – Multiple records per task for short running tasks (like most CICS transactions) cut at task end
  - – Multiple records per SMF interval for long running tasks, like channels or batch jobs.
- The MVS logger has been shown to be very helpful wen generating a lot of SMF data.

Processing the MQ SMF Data

- CSQ4SMFD - Delivered with MQ
  - ▶ Dump format of all records

```
 Class 3 accounting - Queue specific accounting data
 --W-Q-------H-E-X---P-R-I-N-T----
 Address  = 13F285F0
 00000000 : F7020AE0 E6D8E2E3 00000006 5A0247F0 <7..\WQST....!..0>
 00000010 : C95B78F9 DFA95022 5A08CD28 5A078C80 <I$.9.z&.!...!...>
 00000020 : E6D4D8C6 E3C54BC3 F9F5C3F2 F6F9C3F3 <WMQFTE.C95C269C3>
 00000030 : C3F3C2F8 F6C3F040 40404040 40404040 <C3B86C0         >
 00000040 : 40404040 40404040 40404040 40404040 <                >
 00000050 : E6D4D8C6 E3C54BC3 F9F5C3F2 F6F9C3F3 <WMQFTE.C95C269C3>
 00000060 : C3F3C2F8 F6C3F040 40404040 40404040 <C3B86C0         >
 00000070 : 40404040 40404040 40404040 40404040 <                >
 00000080 : C95C269C 798A1F7A F9F5C3F2 F6F9C3F3 <I*..'..:95C269C3>
 00000090 : 00000001 00000000 00000000 00000000 <................>
```

CSQ4SMFD - Notes

N

O

T

E

S

- This is a simple dump format of the SMF data
- Ugly and not typically useful, except if another processing program is not generating a particular field.

Processing the MQ SMF Data

- SupportPac MP1B
  - ▶ The new version is substantially different than the old version
    - There were four report programs delivered in the older version:
      - – MQ1150 – report on the SMF115 data
      - – MQ1160 – report on the SMF116 class 1 data
      - – MQ116S – report on the SMF116 class 3 data
      - – MQCSMF – report on all the SMF data, column formatted data produced
    - There is a single report program delivered in the V7.1 version
      - – MQSMF – report on all the MQ SMF data in both standard report form and CSV files for most of the record subtypes
      - – The report files for the SMF115 data look much like the sections that were part of the MQ1150 report from earlier versions.

Processing the MQ SMF Data

- SupportPac MP1B

  - ▶ There is new information extracted by the new report generator
    - For example, SMDS information
  - ▶ There is information missing from the new reports
    - This has been reported, but will take time to fix
    - For example, the CSV file for the buffer manager does not include the Short on storage, DWT and DMC fields

  - ▶ If you have the older version, keep it
    - If you do not, ask me and I can provide it

MQ Accounting – Task Identification Information

- Uniquely identifies the task

- Includes global task resource use – not individual queue information, which comes next.

MQ Accounting – Task Identification Information

- Example from the MQ116S report program (V7.0.1)

```
z/OS:MPX1  MQ QMGR:QML3  Time: 2013266 08:44:07.50  Jobname:CTSTOR01  Userid:STCRACF
          ====> New task record found     <==========
== Thread type............> CICS
== Connection name........> CTSTOR01
== Operator ID............> STCRACF
== User ID................> CICSUSER
== Channel name...........>
== Chl connection.........>
== Correlator ID..........>       L{QPU2
== Correlator ID.....(HEX)> 152DD3C0D8D7E4F20036934C
== Context token..........>
== Context token.....(HEX)> 000000000000000000000000000000000000
== NID....................>       CTSTOR01ö -°
== NID..............(HEX)> C3E3E2E3D6D9F0F1CC01CA900077E2E8
== Accounting token.......>
== Accounting token..(HEX)> 0000000000000000000000000000000000000000000000
== UOW identifier.........>                    ö -°
== UOW identifier....(HEX)> 4040404040404040404040404040CC01CA9001060001
== Task token : 19-09-2013 13:40:54.14,  5608D3F0,  55FFE000
== Interval   : START 19-09-2013 13:49:38.51
== Interval   : END   23-09-2013 12:44:07.50
== Number of queue blocks for this task        3
```

MQ Accounting – Task Identification Information – cont'd

- Example from the MQ116S report program (V7.0.1) part 2

```
==/
== Task token : 19-09-2013 13:40:54.14,  5608D3F0,  55FFE000
== Interval   : START 19-09-2013 13:49:38.51
== Interval   : END   23-09-2013 12:44:07.50
== Number of queue blocks for this task       3
== Other reqs : Count        1,  Avg elapsed      28,  Avg CPU       13
== Total CPU time under SRB:   0.000267
== Pages     : New          0,  Old          4
WTASVER 5
```

MQ Accounting – Task Identification Information Notes

N O T E S

- The 'interesting bits' of the SMF 116 task identification block includes:
  - Thread type
    - CICS
    - IBM
    - RRSBATCH
    - MOVE – for CHINIT SMF116 data
  - Connection name
    - Jobname
  - Operator and User ID
  - Correlator ID (NOT correlation ID)
    - The CICS transaction can be found here, along with the task number 0036934c
    - The channel name
  - UOW identifier
  - Number of queue blocks that were included in this SMF data
  - Other requests that may have been inflight, but were not captured as part of this SMF record.

MQ Accounting – Task Identification Information

- Example from the MQSMF report program (V7.1)

```
Task statistics

        2 MPX1,QML3,2013/09/23,08:44:07,VRM:710,
        2 QML3 CICS CTSTOR01 opid:STCRACF  userid:CICSUSER Tran:QPU2 task:0036934c
        2 Start time Sep 19 09:40:30 2013 Started in a different time interval
        2 Interval   Sep 19 09:49:14 2013 - Sep 23 08:43:43 2013 : 341668.984234 seconds
        2 == SRB CPU time used    0.000267 Seconds
        2 Other reqs : Total ET          0.000020 Seconds
        2 Other reqs : Total CPU         0.000013 Seconds
        2 Commit count                   0
        2 Commit avg elapsed time        0 uS
```

MQ Accounting – Task Identification Information Notes

N

O

T

E

S

- The MQSMF program formats and prints the task identification section in a more concise way. The example shown if for the same transaction.
- Note that there is no count of queue blocks shown in this report, though the queue information is printed.

MQ Accounting – Queue Information

- Example form the MQ116S SMF report program (V7.0.1)

```
Open name TEAM13.QPU2.STATUS.QUEUE                      Object type:Local Queue
Base name TEAM13.QPU2.STATUS.QUEUE                      Base type  :Queue
Queue indexed by NONE
First opened 23-09-2013 12:44:07.49
Last closed  09-05-2020 22:44:41.19
Page set ID          0,  Buffer pool          0
Current opens        0,  Total requests       3
Generated messages :     0
Persistent messages: GETs         0,  PUTs         0,  PUT1s      0
Put to waiting getter: PUT        1,  PUT1         0
PUTs: Valid      1,  Max size      513,  Min size      513,  Total bytes    513
-MQ call-      N       ET       CT      Susp      LOGW      PSET Epages   skip expire
 Open  :       1       18       18        0
 Close :       1        2        2        0
 Put   :       1       23       23        0         0
Maximum depth encountered       0
```

© 2013 IBM Corporation

MQ Accounting – Queue Information

- Example form the MQSMF report program (V7.1)

```
2 Open name                             TEAM13.QPU2.STATUS.QUEUE
2 Queue type:QLocal                     TEAM13.QPU2.STATUS.QUEUE
2 Page set ID                    0      TEAM13.QPU2.STATUS.QUEUE
2 Buffer pool                    0      TEAM13.QPU2.STATUS.QUEUE
2 Put count                      1      TEAM13.QPU2.STATUS.QUEUE
2 Put avg elapsed time        23 uS     TEAM13.QPU2.STATUS.QUEUE
2 Put avg CPU time            23 uS     TEAM13.QPU2.STATUS.QUEUE
2 Total Queue elapsed time    45 uS     TEAM13.QPU2.STATUS.QUEUE
2 Total Queue CPU used        45 uS     TEAM13.QPU2.STATUS.QUEUE
```

© 2013 IBM Corporation

10

MQ Accounting – Queue Information Notes

N

O

T

E

S

- The queue name, both the open name and base name, are reported.
- The older report explicitly gives the index defined on the queue, even if one is not defined.
  - The new report only includes an index notation when the queue is indexed.
- The pageset and bufferpool ID are given
  - Note there is a known problem here, both are reporting as '0' even though 4 is actually used.
  - Talking with change team, should have an APAR number soon
- Generated messages – gives the number of trigger or event messages generated by the queue activity reported in this record. The new report only shows this if the action occurred.
- Persistent messages – this shows whether persistence is in use. Again, only reported in the MQSMF reports when the messages are persistent.
- Put to waiting getter indicates that there was an application with an empty buffer that was able to receive the message that was put. Technically this should also be reported in the MQSMF V7.1 program. That is under investigation.
- ,.

MQ Accounting – Queue Information Notes

N

O

T

E

S

- The CPU and elapsed time is shown independently for each verb on the MQ116S report. The MQSMF report shows a total, then breaks out the times for the GETs.
- The size of the messages is reported in-stream in the MQ116S report. It is reported in the QSUML output (QSUMS for shared queues) of the MQSMF report, but it is given in megabytes.

Problems found with the SMF116 data - what queues are being used and how?

- Helpful because even as a WMQ admin, it may be a challenge to find out where the queues are being used and abused

- Some specific problems:
  - ▶ Non-indexed queues
  - ▶ High volume request/reply queues in same resource pool
  - ▶ Overuse of Temporary dynamic queues

What queues are being used and how?

- Queue Indexing
  - ▶ Messages that are retrieved using an index-able field benefit from being indexed even when the depth is not high.
    - • Message ID
    - • Correlation ID
    - • Token
    - • Group ID
  - ▶ The greater the depth of the queue the greater the benefit.
  - ▶ The SMF116 queue records show when messages are retrieved using a 'known' field

Queue Indexing - Notes

N

O

T

E

S

- Queue indexing is unique to WMQ on z/OS
- The use of a proper index can substantially improve performance an CPU consumption, as will be shown
- Anecdotally, we've heard of it making a difference when queue depths were as low as 5 on a busy system
- Often the first report of a problem is when there has been a slowdown elsewhere and queue depths have grown unexpectedly

---

Non-Indexed Queue retrieval

```
Open name TEAMXX.NON.INDEXED                      Object type:Local Queue
Base name TEAMXX.NON.INDEXED                       Base type :Queue
Queue indexed by NONE
First opened 12-03-2012 15:12:58.55
Last closed  **-**-**** **:**:**.**
Page set ID          4,  Buffer pool          3
Current opens        1,  Total requests        61
Generated messages :         0
Persistent messages: GETs         0,  PUTs         0,  PUT1s         0
Put to waiting getter: PUT         0,  PUT1         0
GETs: Valid      20,  Max size      80,  Min size      80,  Total bytes   2240
GETs: Dest-S     20,  Dest-G        0,  Brow-S        0,  Brow-G        0, Su cessful destructive      20
Time on queue : Max 4583.730054,  Min  257.434901, Avg 3958.326341
-MQ call-       N         ET         CT      Susp      LOGW      PSET Epages  skip expire
 Get    :      20        384        369         0         0         0    0    3505      0
 Inquire:      20         22         21
Maximum depth encountered          258
```

Non-Indexed Queue retrieval - Notes

N

O

T

E

S

- In the SMF record, the fields of interest are:
  - The Queue Indexing
  - The Type of GET request being made, those with a '-S' are for specific messages (Get by correlid, get by message id, etc.)
  - The total CPU expenditure for the successful gets – the 'CT' column highlighted
  - The number of pages skipped while finding matching messages

---

Indexed Queue Retrieval

```
Open name TEAMXX.INDEXED                          Object type:Local Queue
Base name TEAMXX.INDEXED                          Base type  :Queue
Queue indexed by CORREL_ID
First opened 12-03-2012 15:16:01.44
Last closed  12-03-2012 15:16:50.35
Page set ID           4,  Buffer pool         3
Current opens         0,  Total requests     59
Generated messages :      0
Persistent messages: GETs       0,  PUTs        0,  PUT1s        0
Put to waiting getter: PUT       0,  PUT1        0
GETs: Valid      27,  Max size      80,  Min size      80,  Total bytes    2160
GETs: Dest-S     27,  Dest-G        0,  Brow-S        0,  Brow-G        0, Su cessful destructive    27
Time on queue : Max 4780.946117,  Min  422.046309, Avg 4200.437716
-MQ call-         N       ET       CT       Susp       LOGW      PSET Epages   skip expire
  Get   :        27      105       99        0          0         0     0      0       0
  Inquire:       26       21       20
Maximum depth encountered      250
```

Indexed Queue retrieval - Notes

N

O

T

E

S

- In the SMF record, the fields of interest are:
  - The Queue Indexing
  - The Type of GET request being made, those with a '-S' are for specific messages (Get by correlid, get by message id, etc.)
  - The total CPU expenditure for the successful gets – the 'CT' column highlighted
  - The number of pages skipped while finding matching messages

---

Indexed vs Non - comparison

- Comparing the CPU time, both queues with the same max message depth:
  - Indexed - 27 messages at an average of 99 CPU microseconds
    - 2673 µs for 27 messages retrieved
  - Non-indexed 28 messages at an average 369 CPU microseconds
    - 9963 µs for 27 messages retrieved
  - Difference 272%

- Comparing the elapsed time
  - Indexed - 27 messages at an average 105 microseconds
    - 2835 µs elapsed time for the messages
  - Non-Indexed 28 messages at an average 384 microseconds
    - 10368 µs elapsed time for 27 messages
  - Difference 252%

What queues are being used and how?

- Overuse of Temporary dynamic queues
  - ▶ Often used for responses on both RYO and traditional monitoring tools
  - ▶ All queues created will be in the same resource pool
  - ▶ Quite expensive in CPU

- Temp dynamic queues are identifiable by their name
  - ▶ For example for the MQExplorer uses temporary dynamic queues. The name looks like this
    AMQ.MQEXPLORER.1363497285

Temporary Dynamic Queues

```
Open name TEAMXX.MODEL                          Object type:Local Queue
Base name AMQ.C9422A60F4386075                   Base type  :Queue
Queue indexed by NONE
First opened 12-03-2012 21:24:16.34
Last closed  23-09-2019 17:52:14.24
Page set ID          0,  Buffer pool          0
Current opens        0,  Total requests       10
Generated messages :      0
Persistent messages: GETs        0,  PUTs        0,  PUT1s        0
Put to waiting getter: PUT        0,  PUT1        0
PUTs: Valid       3,  Max size        9,  Min size        9,  Total bytes     27
 -MQ call-        N      ET      CT      Susp      LOGW      PSET Epages   skip expire
  Open  :        1     850     125      727
  Close :        1     113     111        0
  Put   :        3     106     104        0         0
  Inquire:       5      17      17
Maximum depth encountered        3
```

Permanent Queues

```
     -
== Task token : 12-03-2012 21:24:23.42,  55FE03F0,  55FD0000

 Open name TEAMXX.NOT.TEMP                          Object type:Local Queue
 Base name TEAMXX.NOT.TEMP                          Base type  :Queue
 Queue indexed by NONE
 First opened 12-03-2012 21:25:09.23
 Last closed  10-10-2019 00:31:46.22
 Page set ID          0,  Buffer pool        0
 Current opens        0,  Total requests       10
 Generated messages :      0
 Persistent messages: GETs        0,  PUTs        0,  PUT1s        0
 Put to waiting getter: PUT       0,  PUT1       0
 PUTs: Valid      3,  Max size       9, Min size        9,  Total bytes    27
 -MQ call-        N       ET       CT      Susp      LOGW     PSET Epages   skip expire
  Open  :        1       39       38        0
  Close :        1       26       26        0
  Put   :        3      115      113        0         0
  Inquire:       5       18       18
 Maximum depth encountered        3
```

© 2013 IBM Corporation

WMQ Application Performance - Queues

- Choose the right queue:
  - ▶ On z/OS Temporary Dynamic queues should be avoided
    - Higher CPU costs
    - Elapsed time can be significantly longer

- The CPU cost comparison

Verb	TDQ	Permanent	Difference
▶ Open	125	38	238%
▶ Close	111	26	327%
▶ Put	104	113	-8%
▶ Inquire	17	18	-5%

- The Elapsed Time comparison

Verb	TDQ	Permanent	Difference
▶ Open	850	39	2079%
▶ Close	113	26	3347%
▶ Put	106	115	-8%
▶ Inquire	17	18	-5%

© 2013 IBM Corporation

WMQ Application Performance - Queues

- Choose the right queue - continued:

- The Suspend count comparison

Verb	TDQ	Permanent	Difference
Open	727	0	WOW!!!
Close	0	0	0
Put	0	0	0

Using SMF116 to check WMQ Application Performance – Choose the right verbs

- Like any other subsystem, the choice of verbs can improve performance and scalability.
  - Recycling code is a positive
    - Reduces development time and effort
    - Often enforces best practices
    - Can reduce testing time
  - Recycling code is a negative
    - Can introduce performance problems if code not well understood
  - Increased use of a transaction can expose underlying issues

Effect of MQPUT1

- Each MQPUT1:
  - ▶ 117 ųs CPU, for a total 351,000 ųs
  - ▶ 121 ųs Elapsed time, for a total of 363,000

```
PUTs: Valid      3000,  Max size        80,  Min size       80,  Total bytes 240000
-MQ call-          N         ET      CT      Susp        LOGW       PSET Epages   skip expire
 Put1  :         3000       121     117        0          0
```

- Each MQPUT:
  - ▶ 72 ųs CPU, for a total of 216,000 ųs
  - ▶ 74 ųs Elapsed time, for a total of 222,000 ųs

```
PUTs: Valid      3000,  Max size        80,  Min size       80,  Total bytes 240000
-MQ call-          N         ET      CT      Susp        LOGW       PSET Epages   skip expire
 Open  :           1         84      81        0
 Close :           1         18      18        0
 Put   :         3000        74      72        0          0
Maximum depth encountered        6000
```

© 2013 IBM Corporation

Effect of MQPUT1 - Notes

- Remember that both elapsed time and CPU time reported in this section of the MQ116S report is the average time, not the total time.

© 2013 IBM Corporation

19

Effect of MQPUT1

- For one PUT it is less expensive to use an MQPUT1
  - ▶ MQPUT1 - 117 total µs
  - ▶ MQPUT - 171 total µs

- For two PUTs it is less expensive to use an MQOPEN, MQPUT and MQCLOSE
  - ▶ MQPUT1 - 234 total µs
  - ▶ MQPUT -   213 total µs

- Draw your own conclusions

© 2013 IBM Corporation

MQPUT vs MQPUT1 comparison - Notes

N

O

T

E

S

- In one particularly good example of this, ATS was reviewing CPU use for a very high volume queue manager. A single CICS transaction was issuing 7,000+ MQPUT1s to the same queue for each execution. The transaction, once executed a few hundred times a day had become a service. It was now being executed thousands of times a minute.
- Like the Inquisition, no one expected the dramatic jump in CPU.

Capitalware's MQ Technical Conference v2.0.1.3

SMF116 and Long running tasks

- IF the long running task is started after the Class 3 trace
  - SMF 116 records will be cut at each SMF interval and at task end

- If the task is started before the trace is
  - No records are cut
  - APAR PM58798 has been resolved addressing this problem

More information

- There are a number of SupportPacs available:
  - MP16 - Capacity Planning and Tuning for WebSphere MQ for z/OS
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24005907&loc=en_US&cs=utf-8&lang=en
  - MP1H - Performance Report - WebSphere MQ for z/OS V7.1
    - http://www-01.ibm.com/support/docview.wss?uid=swg24031663
  - MP1G - Performance Report - WebSphere MQ for z/OS V7.0.1
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24024589&loc=en_US&cs=utf-8&lang=en
  - MP1F – Performance Report - Performance Report - WebSphere MQ for z/OS V7.0.0
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24020142&loc=en_US&cs=utf-8&lang=en
  - MP1B - Interpreting accounting and statistics data WebSphere MQ for z/OS
    - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24007421&loc=en_US&cs=utf-8&lang=en

Shameless Promotion

Draft Document for Review November 14, 2012 11:02 pm

IBM

SG24-8087-00

WebSphere MQ V7.1 and V7.5 Features and Enhancements

Maximize your investment in WebSphere MQ

Discover new features that bring value to your business

Learn from scenarios with sample configurations

Cezar Aranha
Craig Both
Barry Dearfield
Carolyn Elkins
Alexander Ross
Jamie Squibb
Mark Taylor

ibm.com/redbooks

Redbooks

Questions & Answers