

SQL mot MQ-köer

Lennart Henäng, Folksam

Agenda

- Accessing MQ Queues from SQL
- How to make it happen
- Problems and experiences
- Usefulness

MQ UDFs in DB2

- 19 SQL scalar and table functions
 - READ, RECEIVE, SEND
 - PUBLISH, SUBSCRIBE, and UNSUBSCRIBE
 - VARCHAR, CLOB, or
 - XML, or files (requires XML Extender)
- Two Schemas
 - DB2MQ1C/DMQXML1C – for single phase commit
 - DB2MQ2C/DMQXML2C – for Two-phase Commit
- Runs in a WLM SPAS

Folksam®

3

The functions can be scalar or table functions. For more information on using MQSeries functions, see the information on enabling MQSeries functions in *DB2 Installation Guide* and on programming techniques in *DB2 Application Programming and SQL Guide*.

```
//STEPLIB DD DISP=SHR,DSN=D26.DB2.RUNLIB.LOAD
// DD DISP=SHR,DSN=D26.DB2.DSNEXIT
// DD DISP=SHR,DSN=D26.DB2.DSNLOAD
// DD DISP=SHR,DSN=D25.VERI.SCSQLOAD MQ
// DD DISP=SHR,DSN=D25.VERI.SCSQAUTH MQ
// DD DISP=SHR,DSN=D25.VERI.SCSQANLE MQ
//AMT DD
DISP=SHR,DSN=D36.DB2.SRCLIB.DATA(DSNAMT) AMI
//AMTHOST DD
DISP=SHR,DSN=D36.DB2.SRCLIB.DATA(DSNAMTHT) AMI
//ABNLTERM DD
DISP=SHR,DSN=FOLKSAM.TEST.ABENDAID.DUMPS
```

AMI

- WebSphere MQ Application Messaging Interface (AMI)
- External files with configuration data
 - The AMI Repository (XML) or AMI Cache Files
 - Edited with AMI Tool
- Message - *what*
- Service Point - *where*
 - Logical End Point – DB2.DEFAULT.SERVICE
- Policy - *how*
 - Quality Of Service – DB2.DEFAULT.POLICY
- Service Point and Policy referred to in SQL functions

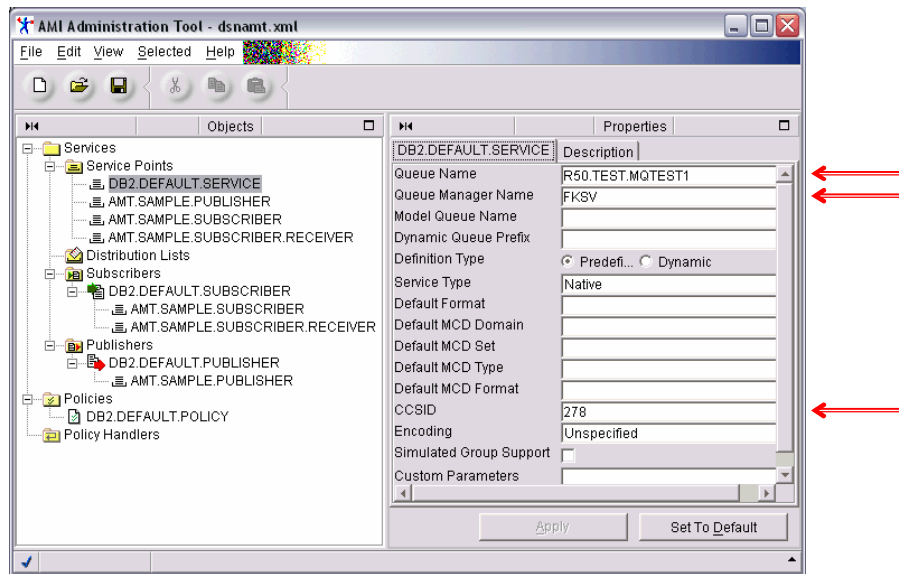
AMI Tool

- AMI is on the 'net, just Google on MA0F
 - ma0f_nt.zip contains AMI Tool and SC34-6065-01
- AMI Tool maintains two XML files in UTF-8
- Configuration files
 - can be edited with ordinary editor supporting UTF-8
 - needs binary file transfer
 - stored in two members of a PDS
 - still in UTF-8
 - can be viewed with Browse (use 'DISPLAY UTF8')
 - can be formatted by AMTBURV program

MA0F is a SupportPac available on
www.ibm.com/software/integration/support/supportpacs/individual/ma0f.html

It includes the AMI Tool and the manual SC34-6065-01.

AMI Tool



Folksam®

6

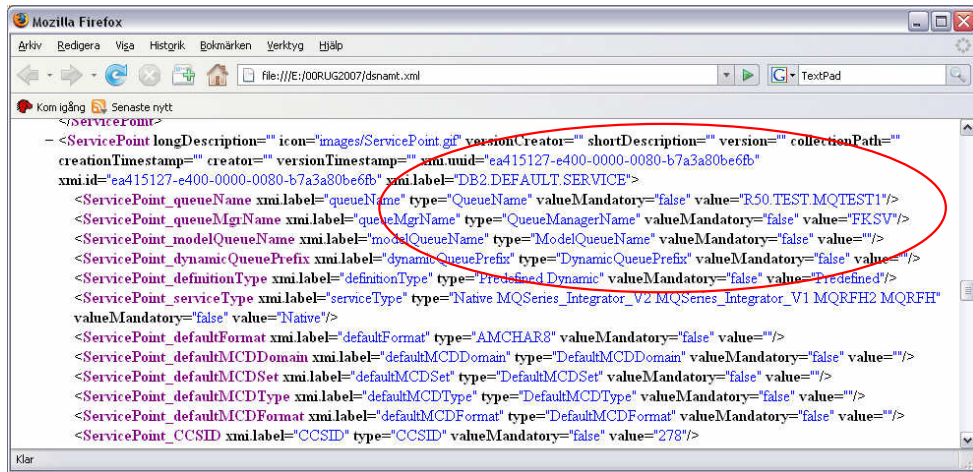
Note the Queue Manager Name and the Queue Name specified to be used for the default service and policy.

XML Files

Two files are needed

- AMTHOST (connection information)
- AMT (queue names etc)

XML Files



```
</?xml>
<?xml:namespace prefix="q" uri="http://schemas.microsoft.com/ServicePoint/2007/ServicePoint" />
<ServicePoint longDescription="" icon="images/ServicePoint.gif" versionCreator="" shortDescription="" version="" collectionPath=""
creationTimestamp="" creator="" versionTimestamp="" xml:uid="ea415127-e400-0000-0080-b7a3a80be6fb"
xml:id="ea415127-e400-0000-0080-b7a3a80be6fb" xml:label="DB2.DEFAULT.SERVICE">
  <ServicePoint_queueName xml:label="queueName" type="QueueName" valueMandatory="false" value="R50.TEST.MQTEST1"/>
  <ServicePoint_queueMgrName xml:label="queueMgrName" type="QueueManagerName" valueMandatory="false" value="FK.SV"/>
  <ServicePoint_modelQueueName xml:label="modelQueueName" type="ModelQueueName" valueMandatory="false" value="" />
  <ServicePoint_dynamicQueuePrefix xml:label="dynamicQueuePrefix" type="DynamicQueuePrefix" valueMandatory="false" value="" />
  <ServicePoint_definitionType xml:label="definitionType" type="PredefinedDynamic" valueMandatory="false" value="Predefined"/>
  <ServicePoint_serviceType xml:label="serviceType" type="Native MQSeries_Integrator_V2 MQSeries_Integrator_V1 MQRFH2 MQRFH"
valueMandatory="false" value="Native"/>
  <ServicePoint_defaultFormat xml:label="defaultFormat" type="AMCHAR8" valueMandatory="false" value="" />
  <ServicePoint_defaultMCDDomain xml:label="defaultMCDDomain" type="DefaultMCDDomain" valueMandatory="false" value="" />
  <ServicePoint_defaultMCDSet xml:label="defaultMCDSet" type="DefaultMCDSet" valueMandatory="false" value="" />
  <ServicePoint_defaultMCDType xml:label="defaultMCDType" type="DefaultMCDType" valueMandatory="false" value="" />
  <ServicePoint_defaultMCDFormat xml:label="defaultMCDFormat" type="DefaultMCDFormat" valueMandatory="false" value="" />
  <ServicePoint_CCSSID xml:label="CCSID" type="CCSID" valueMandatory="false" value="278"/>
</ServicePoint>
</?xml>
```

DB2.DEFAULT.SERVICE maps to queue and queue manager

XML Files

```
Session A - [43 x 80]
Arkiv Redigera Visa Kommunikation Verktyg Fönster Hjälp
Menu Utilities Compilers Help
BROWSE D36.DB2.SRCLIB.DATA(DSNAMT) Converted data shown
Command ==>
***** Top of Data ***** Scroll ==> CSR
<?xml version="1.0" encoding="UTF-8"?>.<!DOCTYPE XML SYSTEM "amt.dtd" >.<!--Change All "MOND" to Name of Queue Manager -->.<XML.xml.version="1.0" timestamp="Wed Nov 29 10:52:32 GMT+01:00 2006" verified="true"/>.<XML.header>.<XML.documentation>.<XML.exporter>.<XML.exporter.name="AMI Administration Tool"/>.</XML.exporter>.<XML.exporterVersion>.<XML.exporterVersion>.</XML.exporterVersion>.</XML.header>.<XML.content>.<Directory longDescription=" icons/images/Directory.gif" versionCreator=" shortDescription=" version=" collectionPath=" creationTimestamp=" creator=" versionTimestamp=" xml.uuid="0d050e0a-0900-0000-0000-09040a0d0000" xml.id="0d050e0a-0900-0000-020b-09040a0d0000"/>.<ServicePointCollection completeSet="false" icon="images/ServicePointCollection.gif" collectionPath=""/>.<ServicePointRef icon="images/ServicePoint.gif" xml:link="simple" xml:label="ServicePoint" type="ServicePoint" href="|ea445127-e400-0000-0000-b7e3a0be6fb" title="DB2.DEFAULT.SERVICE"/>.<ServicePointRef icon="images/ServicePoint.gif" xml:link="simple" xml:label="ServicePoint" type="ServicePoint" href="|020c000a-0900-0000-0000-09040a0d0000" title="AMT.SAMPLE.PUBLISHER"/>.<ServicePointRef icon="images/ServicePoint.gif" xml:link="simple" xml:label="ServicePoint" type="ServicePoint" href="|0406000a-0900-0000-0000-09040a0d0000" title="AMT.SAMPLE.SUBSCRIBER"/>.<ServicePointRef icon="images/ServicePoint.gif" xml:link="simple" xml:label="ServicePoint" type="ServicePoint" href="|0f040f0a-0900-0000-0000-09040a0d0000" title="AMT.SAMPLE.SUBSCRIBER.RECEIVER"/>.</ServicePointCollection>.<DistributionListCollection completeSet="true" icon="images/DistributionListCollection.gif" collectionPath=""/>.<SubscriberCollection completeSet="true" icon="images/SubscriberCollection.gif" collectionPath=""/>.<SubscriberRef icon="images/Subscriber.gif" refType="Subscriber" xml:link="simple" href="|a2ca2c61-e400-0000-0000-b7a3a0be6fb" title="DB2.DEFAULT.SUBSCRIBER"/>.</SubscriberCollection>.<PublisherCollection completeSet="true" icon="images/PublisherCollection.gif" collectionPath=""/>.<PublisherRef icon="images/Publisher.gif" refType="Publisher" xml:link="simple" href="|da172261-e400-0000-0000-b7a3a0be6fb" title="DB2.DEFAULT.PUBLISHER"/>.</PublisherCollection>.</Services>.<PolicyCollection completeSet="false" icon="images/PolicyCollection.gif" collectionPath=""/>.</PolicyCollection>.</XML.content>.</XML>
F10=Left F2=Split F9=Exit F5=Rfind F7=Up F8=Down F9=Swap
F11=Right F12=Cancel F5=Rfind F7=Up F8=Down F9=Swap
04 / 015
Ansluten till fjärrservern/värden mvsprod.intern.folksam.se med lu/pool TCP6042 och p
```

Folksam

This is the DSNAMT "file" shown by Browse after command DISPLAY UTF8.

Using caches for AMI files

- The XML files can be compiled into a load module
 - Run AMTASM10 to produce ASM source

```
//GO EXEC PGM=AMTASM10
//STEPLIB DD DSN=D25.VERI.SCSQLOAD,DISP=SHR
//        DD DSN=D25.VERI.SCSQANLE,DISP=SHR
//AMTHOST DD DSN=D36.DB2.SRCLIB.DATA(DSNAMTHT),DISP=SHR
//AMT     DD DSN=D36.DB2.SRCLIB.DATA(DSNAMT),DISP=SHR
//SYSPRINT DD SYSOUT=*
//ASMHOST DD DSN=D36.DB2.SRCLIB.DATA(AMTHOST),DISP=SHR
//ASMREPOS DD DSN=D36.DB2.SRCLIB.DATA(AMT),DISP=SHR
```

- Assemble and link edit source
- Put load library with cache files in STEPLIB of WLM SPAS

=> Better performance!

AMTBURV to print contents

```
//ALHE03P JOB (D11,D5),LEN,MSGCLASS=Q,CLASS=A,MSGLEVEL=(1,1)
//GEN EXEC PGM=AMTBURV
//STEPLIB DD DSN=DSN=D25.VERI.SCSQLOAD,DISP=SHR
// DD DSN=DSN=D25.VERI.SCSQANLE,DISP=SHR
//* DD DSN=DSN=D36.DB2.AMICACHE,DISP=SHR
//AMTHOST DD DSN=DSN=D36.DB2.SRCLIB.DATA(DSNAMTHT),DISP=SHR
//AMT DD DSN=DSN=D36.DB2.SRCLIB.DATA(DSNAMT),DISP=SHR
//REPORT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

```
Attribute Name is xmi.label, Attribute Value = DB2.DEFAULT.SERVICE
Element Name is ServicePoint_queueName
Attribute Name is xmi.label, Attribute Value = queueName
Attribute Name is value, Attribute Value = R50.TEST.MQTEST1
Element Name is ServicePoint_queueMgrName
Attribute Name is xmi.label, Attribute Value = queueMgrName
Attribute Name is value, Attribute Value = FKSV
```

DB2 Access to Configuration Data

- JCL for WLM SPAS
 - Needs access to MQ load libraries
 - And to configuration data stored in UTF-8

```
//DB2TROUT PROC RGN=OK, APPLENV=DB2TROUT, DB2SSN=DB2T, NUMTCB=20
//IEFPROC EXEC PGM=DSNX9WLM, REGION=&RGN, TIME=NOLIMIT,
// PARM='&DB2SSN, &NUMTCB, &APPLENV'
//STEPLIB DD DISP=SHR, DSN=D26.DB2.RUNLIB.LOAD
// DD DISP=SHR, DSN=D26.DB2.DSNEXIT
// DD DISP=SHR, DSN=D26.DB2.DSNLOAD
//* DD DISP=SHR, DSN=D36.DB2.AMICHACHE
// DD DISP=SHR, DSN=D25.VERI.SCSQLOAD MQ
// DD DISP=SHR, DSN=D25.VERI.SCSQAUTH MQ
// DD DISP=SHR, DSN=D25.VERI.SCSQANLE MQ
//AMT DD DISP=SHR, DSN=D36.DB2.SRCLIB.DATA(DSNAMT) AMI
//AMTHOST DD DISP=SHR, DSN=D36.DB2.SRCLIB.DATA(DSNAMTHT) AMI
```

Register UDFs

- DDL can be found in SDSNSAMP(DSNTIJM1/2)
 - Customize parameters for
 - WLM Application Environment
 - Security – DB2 or USER
 - Encoding Scheme - EBCDIC or ASCII or UNICODE
 - Registers 51 functions
 - E.g., MQSEND has 8 different signatures
 - half of them for VARCHAR(4000)
 - half of them for CLOB(1M)
 - Includes undocumented function GETCOL

IVP for MQ UDFs

- IVP is available in SDSNSAMP(DSNTEJMQ)
 - Customize parameters for
 - To define (create) sample MQ queue
 - HLQ of MQ libraries
 - Queue name
 - Schema name
 - To choose 1PC or 2PC
- For publish/subscribe functions use
 - DSNTEJSQ and DSNTEJSV
- For MQ XML functions and stored procedures use
 - DSNTIJMX for registering
 - DSNTEJX1, DSNTEJX2, and DSNTEJX3 for IVP

Folksam®

14

```
DEFINE QLOCAL('DB2MQ_DEFAULT_Q') REPLACE +
DESCR('INPUT-OUTPUT') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED)
```

Examples of PUT

- PUT a message on a queue
 - a simple message

```
SELECT DB2MQ1C.MQSEND('TESTING MSG 1') FROM SYSIBM.SYSDUMMY1;
-----+-----+-----+-----+-----+-----+-----
1
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

- a message with a correlation id on a specific queue

```
SELECT DB2MQ1C.MQSEND('DB2.DEFAULT.SERVICE', 'DB2.DEFAULT.POLICY',
                      'MSG WITH CORREL ID', 'CORRELID 1')
                      FROM SYSIBM.SYSDUMMY1;
-----+-----+-----+-----+-----+-----+-----
1
```

Examples of GET

- GET a message from a queue
 - the first message on the queue

```
SELECT DB2MQ1C.MQRECEIVE() FROM SYSIBM.SYSDUMMY1;
-----+-----+-----+-----+-----+-----+
TESTING MSG 1
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

- a message with a correlation id on a specific queue

```
SELECT DB2MQ1C.MQRECEIVE('DB2.DEFAULT.SERVICE','DB2.DEFAULT.POLICY',
                        'CORRELID 1') FROM SYSIBM.SYSDUMMY1;
-----+-----+-----+-----+-----+
MSG WITH CORREL ID
```

MQRECEIVE is a destructive read

Examples of GET

- READ a message on a queue - MQREAD
– all messages on a queue

```
-- After putting some new messages on the queue
-----+-----+-----+-----+
SELECT T.* FROM TABLE (DB2MQ1C.MQREADALL()) T;
-----+-----+-----+-----+
MSG
-----+-----+-----+-----+
TESTING MSG 4
TESTING MSG 5
TESTING MSG 6
MSG WITH CORREL ID 2
DSNE610I NUMBER OF ROWS DISPLAYED IS 4
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```



This is a table function!

17

Another example is to store the message data into a DB2 table by copying the data. This is a two-phase commit stuff.

```
INSERT INTO MESSAGES
```

```
SELECT T.MSG FROM TABLE(DB2MQ2C.MQRECEIVEALL()) T;
```

Another example

```
SELECT DB2MQ1C.MQREAD() AS COMPLETE_MESSAGE FROM SYSIBM.SYSDUMMY1;
```

```
-----+-----+-----+-----+-----+-----+-----  
COMPLETE_MESSAGE  
-----+-----+-----+-----+-----+-----+-----  
ANDERS TESTSON STORGATAN 1 183 88 STORSTAD
```

```
SELECT  SUBSTR(DB2MQ1C.MQREAD(),1,6) AS FIRSTNME,  
        SUBSTR(DB2MQ1C.MQREAD(),8,7) AS LASTNAME,  
        SUBSTR(DB2MQ1C.MQREAD(),16,11) AS ADDRESS,  
        SUBSTR(DB2MQ1C.MQREAD(),28,6) AS ZIPCODE,  
        SUBSTR(DB2MQ1C.MQREAD(),35,8) AS CITY
```

```
FROM SYSIBM.SYSDUMMY1;
```

```
-----+-----+-----+-----+-----+-----+-----  
FIRSTNME  LASTNAME  ADDRESS      ZIPCODE  CITY  
-----+-----+-----+-----+-----+-----+-----  
ANDERS    TESTSON   STORGATAN 1 183 88  STORSTAD
```

Felkoder

```
DSNT408I SQLCODE = -443, ERROR:  ROUTINE MQREAD (SPECIFIC NAME MQRD0) HAS
      RETURNED AN ERROR SQLSTATE WITH DIAGNOSTIC TEXT rc=49:amReadMsg
DSNT418I SQLSTATE   = 38H05 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXRRTN SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -818 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFFCCE' X'00000000' X'00000000' X'FFFFFFF'
      X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
DSNE610I NUMBER OF ROWS DISPLAYED IS 0
```

SQLSTATE 38H05 MQSeries Application Messaging Interface failed to read/receive a message.

rc=49:amReadMsg

amReadMsg is not a documented function in the manual

AMRC (Reason Code) 49 = AMRC_TRANSPORT_ERR

/V WLM,APPLENV=DB2TROUT,REFRESH and it works again

Challenges

- Feels a little bit instable, need more time
- AMI deprecated since 2004
 - but DB2s usage is supported by the WMQ lab
- AMI modules are removed from MQ load library
 - can be found on the 'net
 - or be copied from older load libraries
- DB2 will deliver an MQI version on DB2 9
 - retrofit to V8?
 - how to migrate (different parameters needed)?
 - will work in parallel with old implementation

DB2 MQListener

- MQListener monitors a queue and calls a stored procedure with data from message
- Use DSNTIJML to install MQListener
 - Creates SYSMQL.LISTENERS
- db2mqIn1 or db2mqIn2 used to manage MQListener
 - Associates a queue to an SP
 - Specifies number of threads and monitoring frequency
- SP interface
 - schema.proc(in inMsg inMsgType, out outMsg outMsgType)
 - outMsg can be put on reply-to queue
- IVP in DSNTEJML and DSNTEJSP

Folksam®

21

The data type for *inMsgType* and the data type for *outMsgType* can be VARCHAR, CLOB, or BLOB of any length and are determined at startup. The input data type and output data type can be different data types. If an incoming message is a request and has a specified reply-to queue, the message in outMsg will be sent to the specified queue.

```
CREATE PROCEDURE TEST.APROC (  
  IN PIN VARCHAR(25),  
  OUT POUT VARCHAR(2))  
LANGUAGE SQL FENCED NOT DETERMINISTIC NO DBINFO COLLID  
TESTLSRN WLM ENVIRONMENT TESTWLMX  
ASUTIME NO LIMIT STAY RESIDENT NO PROGRAM TYPE MAIN  
SECURITY USER  
PROCEDURE1: BEGIN INSERT INTO PROCTABLE VALUES(PIN); SET  
POUT = 'OK';  
END PROCEDURE1
```

More Information

- DB2 SQL Reference
- DB2 Application Programming and SQL Guide, Chapter 33
- DB2 Installation Guide, Chapter 7

Summary

- MQ SQL UDFs can be useful for SQL-programmers
- We can use existing infrastructure to access MQ queues
- We did not test
 - Two-phase commit version of UDFs
 - XML UDFs and Stored Procedures
 - Needs XML Extender
 - Performance
 - DB2 MQListener - yet
 - Corresponding functionality in DB2 LUW