SupportPac MA1O for Websphere MQ for z/OS

Rexx Application Trigger Monitor for TSO

Author: Willi Jorg
Owner: Alfred Frischmann

Document History

Document Location

This is a snapshot of an on-line document. Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

Revision History

Date of this revision: 05.03.2009	Date of next revision n/a

Revision Number	Revision Date	Summary of Changes	Changes marked
(1)	05.03.2009	Document created	(N)

Contents

1. Introduction

This document introduces a Websphere MQ-Series Application Trigger Monitor for TSO. The Monitor is a TSO job which starts a program written in Rexx. It monitors the specified initiation queue of the specified MQ-Manager and starts the associated process.

The specific characteristic of this monitor is the ease of use and the minimum of prerequisites to get the monitor run. The activation of the monitor is quite simple

- Just copy the software components in the appropriate libraries
- Customize the MQ-manager and the initiation queue in the job card
- Do the MQ-Definitions for the MQ-Manager, like
    - Initiation queue
    - Process
    - Application queue
- Submit the TSO job

Then your WMQ-Environment is ready to handle incoming messages in the application queue using the Application Trigger Monitor.

2. Application Trigger Monitor in Rexx

2.1 Concept of WMQ Trigger Mechanism

Just for recapitulation find below a brief description of the trigger mechanism of WMQ.

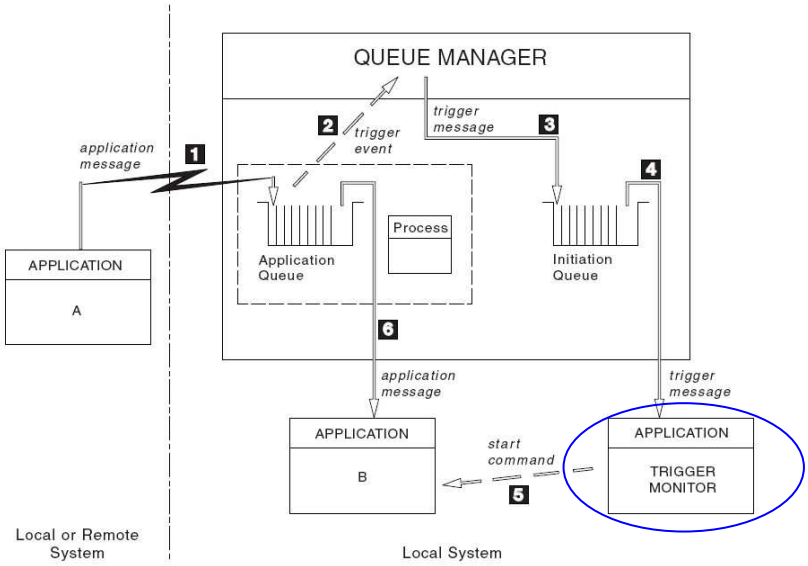

Figure 1: Standard scenario of the WMQ trigger mechanism

1. Application A puts a message into Application Queue
2. Queue Manager checks due to definition of Application Queue whether there is to create a Trigger.
3. Queue Manager puts a Trigger Message into Initiation Queue, which belongs to Application Queue
4. Trigger Monitor gets the Message from Initiation Queue.
5. Trigger Monitor starts Application B, which is defined via process that is associated with the application queue.
6. Application B gets Message from Application Queue.

2.2 Example of WMQ definitions

Applicaton A:	ESIMQCR on z/VM (MCEVM2)
Queue Manager:	MQS
Application Queue:	ACCP.BILLING.ESI.DEVLP
Initiation Queue:	MCEVS1.INITIATION.DEVLP
Application Trigger Monitor:	Job MQMONPDJ
Process Name:	ACCPDATA.BILLING.ESI.DEVLP.PROCESS
Application ID (Application B):	TSSDEVP.SOURCES.ACCPADM.EXEC(STESIJOB) on z/OS (MCEVS1)
Environment data:	SOURCES.ACCPADM.JOBS(MQESIFCR)

2.3 Software components

The software components are listed in the figure above, where you see a browse screenshot of the example libraries.

```
BROWSE            TSSDEVP.TRIGMON.SOURCE                  Row 00001 of 00003
           Name     Prompt       Size   Created          Changed          ID
_________ DEPLOYER                116  2008/12/10  2008/12/10 06:04:23  FRISCHM
_________ MQMONIPD                 25  2008/11/24  2008/12/01 07:57:16  FRISCHM
_________ MQMONJOB                 13  2008/10/10  2009/01/24 11:53:10  FRISCHM
          **End**

BROWSE            TSSDEVP.UTILITY                         Row 00001 of 00003
           Name     Prompt       Size   Created          Changed          ID
_________ LOG2SAVE                100  2009/02/17  2009/02/18 08:05:49  FRISCHM
_________ MQTILITY                427  2008/11/23  2009/01/24 12:22:25  FRISCHM
_________ SAVE2LOG                 88  2009/01/23  2009/02/04 13:07:06  FRISCHM
          **End**
```

Remark:

- Prerequisite for usage of SAVE2LOG (called in MQTILITY) is a “PIPE Environment”.
- Alternatively you can change SAVE2LOG in LOG2SAVE which uses EXECIO instead PIPE

2.3.1 TSO job MQMONPDJ

The Trigger Monitor is embedded in a TSO job named MQMONPDJ. The job calls the Rexx program MQMONIPD.

The association to MQ-Manager is done via parameter of the Rexx program.

In example below we have

- Queue-Manager: MQS
- Initiation Queue: MCEVS1.INITIATION.DEVLP

A library named &HOME..UTILITY is mandatory.

```
//MQMONPDJ JOB (7949,SYS),'TSSDEVP',
//             NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
// SET HOME=TSSDEVP
//*-------------------------------------------------------
//* REXX APPLICATION TRIGGER MONITOR FOR TSO V1.0
//* PROGRAM  : MQMONIPD
//* PARAMETER: QUEUE-MANAGER   : MQS
//*            INITIATIAN QUEUE: MCEVS1.INITIATION.DEVLP
//*-------------------------------------------------------
//STEP1   EXEC PGM=IKJEFT01,REGION=64M
//SYSPRINT DD SYSOUT=*
//SYSEXEC  DD DSN=&HOME..TRIGMON.SOURCE,DISP=SHR
//         DD DSN=&HOME..UTILITY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 MQMONIPD MQS(MCEVS1.INITIATION.DEVLP)
/*
```

Customization must be done here to refer to the characteristics of your environment, such as

- Name of Queue-Manager
- Name of Initiation Queue
- Name for Source Libraries

2.3.2 Rexx programs

2.3.2.1 MQMONIPD

This Rexx program gets and sets parameters for the monitoring functionality of MQTITLIY and calls MQTILITY.

```
/*REXX**-------------------------------------------------------------*/
/*   MQMONIPD EXEC                                                   */
/*                                                                   */
/* yyyy-mm-dd  who what----------------------------------------     */
/* 2008-11-15  AFR Init                                              */
/* 2008-12-01  AFR Version2                                          */
/*                                                                   */
/*-------------------------------------------------------------------*/
Parse Arg argus .
Parse Value SPACE(TRANSLATE(argus,,"(')"),1) With $qm $qn .
QManager     = $qm
Qname        = $qn
/*---------------------------------------------------------*/
/*          FuncName(MESSAGE/MONITOR)                      */
/*          |       Maxreclen   Default 1024               */
/*          |       |    WaitSeconds oo=NeverEndingWait    */
/*          |       |    |  RetryCounter "*"=0             */
/*          |       |    |  | OFF/KEEP: delete/keep data   */
/*          v       v    v  v v          on Queue          */
/*---------------------------------------------------------*/
PutGetOpts="MONITOR 2048 oo * OFF"
$RC=MQtility("GET",PutGetOpts,Qname)
say $RC
Parse Value $RC With $RC $result
```

Whereby the parameters for MQTILITY are understood as follows:

- MONITOR: MQTILITY() goes into a never ending “GET” loop
- 2048: explicit data-record-length of WMQ-message
- oo: never-ending-symbol for get-loop above
- *: Retry-counter for MQ function (*== NoRetry)
- OFF: run MqCMIT to remove data from queue

2.3.2.2 MQTILITY

This program serves as MQ-Function-Tool for Rexx applications. It contains all required basics for MQ-Messaging as:

- MqInit
- MqOpen
- MqPut
- MqGet
- MqDisc
- MqTerm
- and special sub-function as MqMonitor

Each Rexx-Program may write a record into a MQ-Queue and read a record from a MQ-Queue without any “knowledge” about MQ-data-processing. For example:

- ProgA calls MQTILITY("PUT","MESSAGE",Qname,,outrec) to write one record into a MQ-Queue.
- ProgB calls MQTILITY(“GET”,”MESSAGE”,Qname,UniqueTicketNmbr) to get a record belonging to unique ticket number from a MQ-Queue, whereby ticket-number is a synonym for MQ-CorrelationID

Using MQTILITY a MQ-Monitor program simply consists of about two lines of code (see above).

Remark:

The program is delivered using program SAVE2LOG for logging the program calls.

If there is no BATCHPIPES Environment available use LOG2SAVE instead SAVE2LOG for logging. Just customize your MQTILITY program by do a global change from SAVE2LOG to LOG2SAVE.

2.3.2.3 DEPLOYER

This program determines the first level qualifier of the mandatory library named &HOME..UTILITY and does various dataset allocations. The name UTILITY is fix.

2.3.2.4 SAVE2LOG

This program logs the start and end of any program started due to a trigger definition for the monitored initiation queue. The program uses Batchpipes for handling of the log dataset.

Any log entry consists of program name and parameter values in case of start and return code when ended.

The standard name of the log dataset is &HOME..SAVE2LOG.MQTILITY.

2.3.2.5 LOG2SAVE2

This program logs the start and end of any program started due to a trigger definition for the monitored initiation queue. It is a clone of SAVE2LOG using EXECIO instead of Batchpipes for handling of the log dataset.

Any log entry consists of program name and parameter values in case of start and return code when ended.

The standard name of the log dataset is &HOME..LOG2SAVE.MQTILITY.

2.4 Build Environment

The described application monitor was tested using follow system software and middleware

- z/OS 1.7 or higher
- WebSphere MQ for z/OS V7 or higher
- WMQ-SupportPac MA18 or MA95 – A REXX interface to Websphere MQ
- IBM Library for REXX/370
- BATCHPIPES FOR OS/390, 5655-D45 02.01.00
  (optional, but prerequisite when using SAVE2LOG instead LOG2SAVE)

2.5 Trigger Monitor – Step by Step

In this chapter the parameter flow and the software architecture is shown starting with the WMQ-Definitions and ending up in the Monitor program.

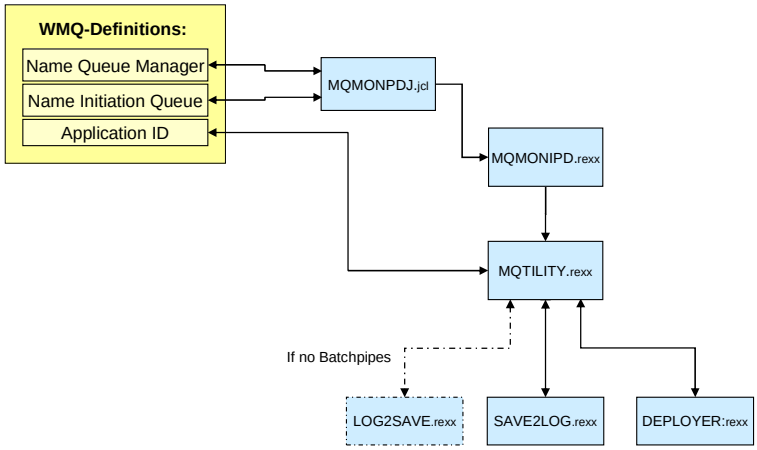

Figure 2: Software architecture

1. WMQ Definitions

```
MQ-Setup-QUEUE:
 Queue name . .   : MCEVS1.INITIATION.DEVLP
 Disposition . . : QMGR    MQS
MQ-Setup-PROCESS:
 Application ID   : TSSDEVP.SOURCES.ACCPADM.EXEC(STESIJOB)   << ValOf.AID
 User data      . :
 Environment data: SOURCES.ACCPADM.JOBS(MQESIFCR)
```

2. Job MQMONJOB

```
//MQMONPDJ JOB (7949,SYS),'Rexx MQ Appl Trigmon',
//             NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
// SET HOME=TSSDEVP
//*************************************************************
//STEP1   EXEC PGM=IKJEFT01,REGION=64M
//SYSPRINT DD SYSOUT=*
//SYSEXEC  DD DSN=&HOME..TRIGMON.SOURCE,DISP=SHR
//         DD DSN=&HOME..UTILITY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 MQMONIPD MQS(MCEVS1.INITIATION.DEVLP)
/*
```

3. Caller program MQMONIPD

```
PutGetOpts="MONITOR 2048 oo * OFF"
$RC=MQtility("GET",PutGetOpts,Qname)
```

4. Monitor program MQTILITY

```
ValOf.AID=" TSSDEVP.SOURCES.ACCPADM.EXEC STESIJOB"
Parse Value ValOf.AID With $PROGHOME $PROGNAME .

$=DEPLOYER("TSSDEVP.SOURCES.ACCPADM.EXEC")
rc=STESIJOB()
```

3. Appendix

3.1 Example Definitions

3.1.1 Websphere MQ-Definitions

3.1.1.1 Application Queue

```
                          Display a Local Queue - 4

 Press F7 or F8 to see other fields, or Enter to refresh details.

                                                                More:  - +
 Queue name  . . . . . . . . : ACCP.BILLING.ESI.DEVLP
 Disposition . . . . . . . . : QMGR    MQS

 Trigger Definition

    Trigger type . . . . . . : F  F=First, E=Every, D=Depth, N=None
    Trigger set  . . . . . . : Y  Y=Yes, N=No

    Trigger message priority : 0  0 - 9
    Trigger depth  . . . . . : 1          1 - 999999999

    Initiation queue . . . . : MCEVS1.INITIATION.DEVLP
    Process name . . . . . . : ACCPDATA.BILLING.ESI.DEVLP.PROCESS
    Trigger data . . . . . . :
```

3.1.1.2 Process Definition

```
                         Display a Process - 1

Press F8 to see further fields, or Enter to refresh details.

                                                              More:    +
Process name  . . . . . . . . ACCPDATA.BILLING.ESI.DEVLP.PROCESS
Disposition . . . . . . . . . : QMGR    MQS
Description . . . . . . . . . : ACCP-ESI Interface:
                                CREATE ESI-SIF-FILE
Application type  . . . . . . : MVS

Application
   ID . . . : TSSDEVP.SOURCES.ACCPADM.EXEC(STESIJOB)
```

```
                         Display a Process - 2

Press F7 to see previous fields, or Enter to refresh details.

                                                              More:  -
Process name  . . . . . . . . : ACCPDATA.BILLING.ESI.DEVLP.PROCESS
Disposition . . . . . . . . . : QMGR    MQS

User data . . . . . . . . . . :


Environment data  . . . . . . : SOURCES.ACCPADM.JOBS(MQESIFCR)

Last alteration time  . . . . : 2009-02-03 16.01.30
```

3.1.2 Example Application referring to process

```
/*REXX*************************************************************/
/*                                                                 */
/* Name: STESIJOB                                                  */
/*                                                                 */
/* Start Job for Creating ESI-SIF-File                             */
/* The procedure is called by MQ-Trigger Process                   */
/* Following external programs are called:                         */
/*  - DEPLOYER: Bootsstrap to establish access to other libraries  */
/* Following tables are involved:                                  */
/*  - none                                                         */
/*                                                                 */
/*  CREATION START DATE:    2009-02-03                             */
/*-----------------------------------------------------------------*/
/* trace ?R  */
ADDRESS 'TSO'
  Arg MQ_Queue_Name        ,,
      MQ_Process_Name      ,,
      MQ_Trigger_Data      ,,
      MQ_Application_Type  ,,
      MQ_Application_Id    ,,
      MQ_Environment_Data  ,,
      MQ_User_Data         ,,
      MQ_Queue_Manager_Name
rc = '0'
/*-------------------------------------------------*/
/* Determine 'First Level Qualifier' of EXEC-DSN   */
/*-------------------------------------------------*/
utility_dir = ".UTILITY"
Where_Iam = deployer('*' !! utility_dir)
Mode = Left(Where_Iam,7)
/*-------------------------------------------------*/
/* Prepare jobname and submit job                  */
/*-------------------------------------------------*/
IF MQ_Environment_Data <> '' then
  jobname = Strip(mode) !! '.' !! Strip(MQ_Environment_Data)
else
  jobname = Strip(mode) !! '.SOURCES.ACCPADM.JOBS(MQESIFCR)'
jobname_hig = "'"jobname"'"
"submit" jobname_hig
return rc
```

3.1.3 Example log entry

```
BROWSE    TSSDEVP.SAVE2LOG.MQTILITY                  Line 00000043 Col 001 080
20090212-163252 BEFORE START: $RC=STESIJOB('ACCP.BILLING.ESI.DEVLP' ,'ACCPDATA.B
20090212-163252 AFTER  START: 0 OK
```

Remark:

The figure above is truncated at line 80 – just to give an impression.

3.1.4 Example MQ SDSF Entries

```
SDSF DA ESA2     MCEVS1   PAG  0  CPU/L/Z  16/ 16/  0  LINE 1-3 (3)
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     MQMONPDJ STEP1             JOB06017 STCUSER  A LO  FF  550   0.00   0.00
     MQSMSTR  MQSMSTR  PROCSTEP STC03590 STCUSER    NS  FE 2519   0.00   0.00
     MQSCHIN  MQSCHIN  PROCSTEP STC03591 STCUSER    IN  F4 2141   0.00   0.00
```