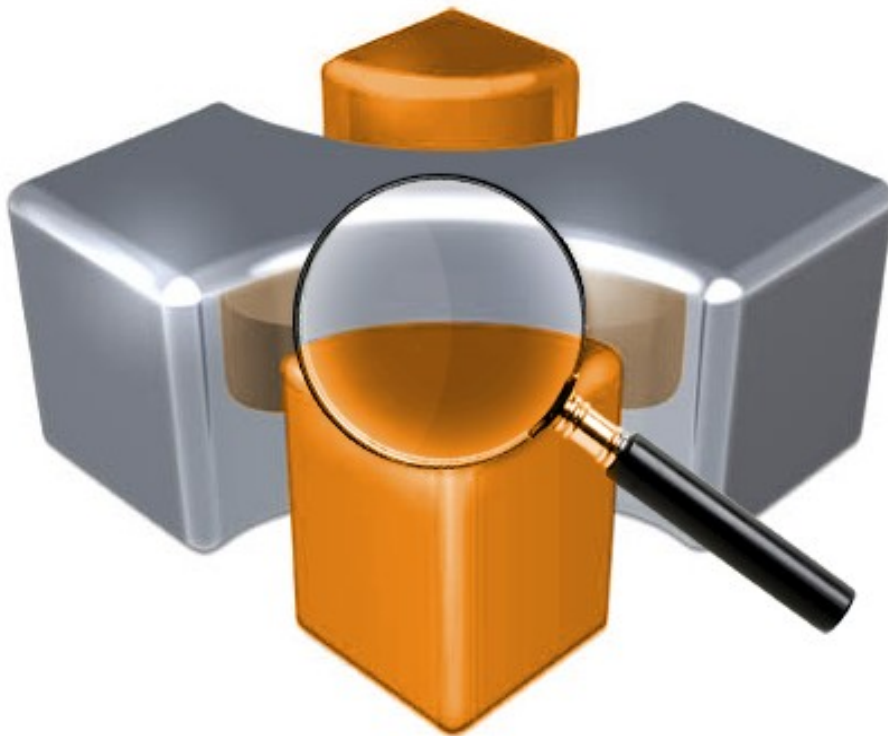


MQ Auditor Overview



Capitalware Inc.
Unit 11, 1673 Richmond Street, PMB524
London, Ontario N6G2N3
Canada
sales@capitalware.com
<https://www.capitalware.com>

Last Updated: January 2021.
© Copyright Capitalware Inc. 2010, 2021.

Table of Contents

- 1 INTRODUCTION.....1
- 1.1 OVERVIEW.....1
- 1.2 EXECUTIVE SUMMARY.....2
- 1.3 PREREQUISITES.....3
 - 1.3.1 *Operating System*.....3
 - 1.3.2 *IBM MQ*.....4
 - 1.3.3 *Windows 32-bit*.....4
 - 1.3.4 *Windows 64-bit*.....4

1 Introduction

1.1 Overview

MQ Auditor (MQA) is a solution that allows a company to audit / track all MQ API calls performed by MQ applications that are connected to a queue manager. The API Exit operates with IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 in Windows, Unix, IBM i and Linux environments.

MQA audits the following MQ API calls:

- MQCONN, MQCONNX, MQOPEN, MQGET, MQPUT, MQPUT1, MQINQ, MQSET, MQCLOSE, MQDISC, MQBACK, MQBEGIN and MQCMIT.
- XASTART, XAEND, XAOPEN, XACLOSE, XACOMMIT, XACOMplete, XAFORGET, XAPREPARE, XARECOVER, XAROLLBACK, AX_REG and AX_UNREG.
- MQCALLBACK, MQCB, MQCTL, MQSTAT, MQSUB and MQSUBRQ.

MQA is designed to provide the user with all of the information of an MQ API call in "human readable" format. Human readable implies that it will convert binary fields into their MQ defined names (i.e. ObjType=MQOT_Q). It will convert Command Server / PCF messages into human readable messages. The user can control the fields that are outputted for each of the following MQ structures: MQCNO, MQOD, MQGMO, MQPMO and MQMD. For MQ v7 & higher, the following fields are outputted: Message Properties, MQCBC, MQCBD, MQSTS and MQSD.

MQA handles the following MQ embedded message types: MQCIH, MQDH, MQDLH, MQIIH, MQRFH, MQRFH2, MQRMH, MQTM, MQWIH, MQXQH, MQHSAP, SMQBAD. The user can control the fields that are outputted for each of the MQ embedded messages.

MQA's default behavior is to log all MQ API calls by all applications (users). To override the default, the user can define 4 types of filters: Applications, UserIDs, Queues and Topics.

- Applications means that MQ Auditor will log all MQ API calls whose application name matches the filter value.
- UserIDs means that MQ Auditor will log all MQ API calls whose UserID matches the filter value.
- Queues means that MQ Auditor will log all MQ API calls whose queue name matches the filter value.
- Topics means that MQ Auditor will log all MQ API calls whose topic name matches the filter value.

MQA is designed to output 1 line per API call (1 long line). The output (audit) information is written to plain text CSV (Comma Separate Value) files. There are 2 Audit CSV file types: QMgr and Queue/Topic. Audit information for MQGET, MQPUT, MQPUT1, MQINQ and MQSET related to a queue is written to the Queue Audit file and audit information for all other calls is written to the QMgr Audit file.

The user can choose to have the output (audit) information written to a local or remote queue rather than to a file. From an MQA point of view, this is a very dangerous feature. The reason it is dangerous is that MQA audits / monitors all MQ API calls and for each call it generates audit data. If the audit data is written to a queue then this action can potentially cause an endless loop (MQA will audit itself, over and over again). Therefore, the "audit queue" is not audited / monitored nor is the transmit queue if remote queue is used. If the user is using a remote queue then it is strongly recommended that a separate channel and transmit queue be used so that the normal transmit queue can be audited / monitored. The user will need a program to read the messages from the 'audit queue' and to write the information somewhere (i.e. database).

Audit Queue Off Load (AQOL) is a new companion application for MQ Auditor. The purpose of AQOL is to retrieve audit records from the audit queue and write the audit records to plain text CSV (Comma Separate Value) files. For more information regarding the AQOL program, please read the ***AQOL Installation and Operation*** manual.

MQA can explicitly not write audit information for particular queues, topics, UserIds and applications. It can also explicitly not write information when MQ issues particular reason codes (i.e. 2033).

On AIX, HP-UX, Linux, Solaris and Windows, MQA can be configured and used with a non-default installation of MQ in a multi-install MQ environment.

Note: Raspberry Pi is a Linux ARM 32-bit OS (Operating System). Hence, simply follow the Linux 32-bit instructions for installing and using the solution on a Raspberry Pi.

1.2 Executive Summary

The MQ Auditor solution contains an API Exit.

The API Exit is available in 3 forms:

- Windows DLL (32-bit & 64-bit)
- Shared library for AIX, HP-UX, Linux, and Solaris.
- IBM i exit module

The major features of MQ Auditor are as follows:

- Audit / track all MQ API calls issued by an MQ application
- MQA is designed to provide the user with all of the information of an MQ API call in "human readable" format.
- MQA will process all standard MQ structures
- MQA will process all standard embedded message structures
- Provides the audit files in an easy to manage CSV (Comma Separated Value) format
- The user can define 4 types of filters: Applications, UserIDs, Queues & Topics
- The user can select to have the CSV data written to a queue rather than a file
- The user can select to not have audit information written for particular reason codes

1.3 Prerequisites

This section provides the minimum supported software levels. These prerequisites apply to server-side installations of MQ Auditor.

1.3.1 Operating System

MQ Auditor can be installed on any of the following supported servers:

1.3.1.1 IBM AIX

- IBM AIX 6L version 6.1 or higher

1.3.1.2 HP-UX IA64

- HP-UX v11.23 or higher

1.3.1.3 IBM i (OS/400)

- IBM i V6R1 or higher

1.3.1.4 Linux x86

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.3.1.5 Linux x86_64 (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.3.1.6 Linux on POWER

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.3.1.7 Linux on zSeries (64-bit)

- Red Hat Enterprise Linux v5, v6, v7, v8
- SUSE Linux Enterprise Server v11, v12, v15

1.3.1.8 Raspberry Pi (Linux ARM 32-bit)

- Raspberry Pi OS v9 or higher

1.3.1.9 Sun Solaris

- Solaris SPARC v10 or higher
- Solaris x86_64 v10 or higher

1.3.1.10 Windows

- Windows 2008, 2012 or 2016 Server (32-bit & 64-bit)
- Windows 7, 8, 8.1 or 10 (32-bit & 64-bit)

1.3.2 IBM MQ

- IBM MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2 (32-bit and 64-bit)

Operating System	MQ v7.1, v7.5, v8.0, v9.0, v9.1 and v9.2
AIX v6.1 or higher	32-bit & 64-bit
HP-UX IA64 v11.23 or higher	32-bit & 64-bit
IBM i (OS/400)	64-bit
Linux x86	32-bit
Linux x86 64	32-bit & 64-bit
Linux on POWER	32-bit & 64-bit
Linux on zSeries	32-bit & 64-bit
Raspberry Pi ARM	32-bit
Solaris SPARC v10 or higher	32-bit & 64-bit
Solaris x86 64 v10 or higher	32-bit & 64-bit
Windows 2008, 2012, 2016, 7, 8, 8.1 & 10	32-bit & 64-bit

1.3.3 Windows 32-bit

The following is the software prerequisite for Windows 32-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
<https://www.microsoft.com/en-ca/download/details.aspx?id=5555>

1.3.4 Windows 64-bit

The following is the software prerequisite for Windows 64-bit:

- Microsoft Visual C++ 2010 Redistributable Package (x64)
<https://www.microsoft.com/en-ca/download/details.aspx?id=14632>