

# **Getting Started with Universal Command Agent for SOA - MQ Connector**

Universal Agent 7.9.x

© 2024 by Stonebranch, Inc. All Rights Reserved.

# Table of Contents

1 Objective.....	5
2 Installation Requirements .....	6
3 Installation.....	7
4 MQ Environment Verification.....	8
5 Running a Universal Command Agent for SOA Job on z/OS Connecting to MQ Connector .....	9
6 Running a Universal Command Agent for SOA Job on UNIX Connecting to MQ Connector .....	11

- [Objective](#)
- [Installation Requirements](#)
- [Installation](#)
- [MQ Environment Verification](#)
- [Running a Universal Command Agent for SOA Job on z/OS Connecting to MQ Connector](#)
- [Running a Universal Command Agent for SOA Job on UNIX Connecting to MQ Connector](#)

# 1 Objective

The objective of this document is to assist in the following activities regarding the Universal Command Agent for SOA: MQ Connector:

- Installing Universal Agent for SOA 7.9.x, which is comprised of:
  - Universal Command Agent for SOA
  - Universal Event Monitor for SOA
- Running Universal Command Agent for SOA with an MQ Connector.

## 2 Installation Requirements

For installation requirements, see [Universal Agent for SOA for UNIX - Installation Requirements](#).

## 3 Installation

For installation information, see [Universal Agent for SOA for Linux Installation](#).

## 4 MQ Environment Verification

Verify that you have a working MQ environment. You must define the following MQ values, as these are needed for the Universal Command Agent for SOA jobs that you will submit: `queuemanager`, `queueName`, and `channel`.

You now can run jobs in MQ using the Universal Command Agent for SOA: MQ Connector.

## 5 Running a Universal Command Agent for SOA Job on z/OS Connecting to MQ Connector

### Step 1

Create the UCMD Manager JCL. This provides the UCMD Manager options, references to the MQ Connector options, and the payload. It has the following format:

```
//XXXXXXXXX JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
000002 //*
000003 //*****
000004 //*MQ queue test for Publish
000005 //*UCMD is the proc that calls UC Manager
000006 //*LOGON is the DD with userid and passwd (can use encrypted)
000007 //*SCR is the script that contains the MQConnector information
000008 //* to connect to an MQ Broker*
000009 //*UNVIN provides the payload for the SCRIPT in SCR*
000010 //*****
000011 //*
000012 //*   JCLLIB ORDER=LIB.V3207.UNV.UNVCONF
000013 //*
000014 //UCMD   EXEC UCMDPRC
000015 //LOGON  DD DISP=SHR,DSN=USER123.UAC.LOGON(USER)
000016 //SCR    DD DISP=SHR,DSN=USER123.UAC.SCR(MQPUB)
000017 //UNVIN  DD DISP=SHR,DSN=USER123.UAC.PYL(MQPYL)
000018 //UNVOUT DD SYSOUT=*
000019 //UNVERR DD SYSOUT=*
000020 //SYSIN  DD *
000021 -s scr
000022 -script_type SERVICE*
000023 -i ucaserver -f logon
```

**Step 2**

Create the MQ Connector Command Options Data Set Member.

This member contains the [UCA for SOA command options](#) for the MQ Connector that specifies the required information to submit a job to the MQ environment. It is referenced with the **SCR** ddname and has the following format:

```
-protocol mq
-mep Publish
-mqhost MQHOST
-mqueueanagername MyQueueManager
-mqueueuename UpsQaQueue
-mqchannel UpsQaChannel
-timeoutsec 120
```

**Note**

If the port on which the MQ Broker is listening has been changed from its default value (1414), you must include the **-mqport** option to specify the current port.

**Step 3**

Create the Payload Data Set Member. This member contains the MQ message and is read in via STDIN.

**Note**

The **LRECL** length depends on the job it describes. Verify that your data set member record length can accommodate the maximum line length of your message.

Example:

```
000001
000002 Hello...this is a payload in an MQ message.
```

## 6 Running a Universal Command Agent for SOA Job on UNIX Connecting to MQ Connector

**Step 1** Create the UCMD script file (**Mqopt**) to contain the the [UCA for SOA command options](#) for the MQ Connector that specifies the required information to submit a job to the MQ environment.

Mqopt contains:

```
-protocol mq
-mep Publish
-mqhost MQHOST
-mqueueanagername MyQueueManager
-mqueueuename UpsQaQueue
-mqchannel UpsQaChannel
-timeoutsec 120
```

**Note**

If the port on which the MQ Broker is listening has been changed from its default value (1414), you must include the `-mqport` option to specify the current port.

**MQPayload.xml**

```
Hello...this is a payload in an MQ message.*
```

**Step 2** From a command prompt, execute the following command to send a message to an MQ Queue:

```
ucmd -script Mqopt -script_type SERVICE -i ucaserver -u user -w user < MQPayload.xml
```

You can also execute the command using the Universal Command options for STDIN (`-I` for input and `-F` for file):

```
ucmd -script Mqopt -script_type SERVICE -i ucaserver -u user -w user -I -F MQPayload.xml
```

**Example output:**

```
MQ message published successfully on destination UpsQaQueue.
```