



**stonebranch**  
you imagine IT. we automate IT.

## **Universal Command Agent for SOA 5.2.0**

### **MQ Connector**

© 2015 by Stonebranch, Inc. All Rights Reserved.

# Getting Started with Universal Command Agent for SOA - MQ Connector 5.2.0

- 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

## Objective

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

- Installing Opswise Universal Agent for SOA 5.2.0, which is comprised of:
  - Universal Command Agent for SOA
  - Universal Event Monitor for SOA
- Running Universal Command Agent for SOA with an MQ Connector.

## Installation Requirements

The following is required for running Universal Command Agent for SOA with an MQ Connector:

- Opswise Universal Agent 5.2.0.0 or later (32-bit package); installed, licensed, and running.
- MQ Environment version 6 or later, with working queues.
- MQ Client jar files for native communication to MQ must be in the following path:

```
/opt/universal/uac/container/webapps/axis2/WEB-INF/lib  
  
com.ibm.mq.commonservices.jar  
com.ibm.mq.jar  
com.ibm.mq.pcf.jar  
com.ibm.mq.headers.jar  
com.ibm.mq.jmqi.jar  
connector.jar
```

The MQ Client for Java version 7.0 package with the latest fix pack is recommended.

When using a MQ CCDT to establish connections to queue managers, 7.0.1.3 or later is highly recommended.

## Installation



### Note

These instructions describe installation of the Opswise Universal Agent for SOA 5.2.0 for AIX package.

Opswise Universal Agent for SOA 5.2.0 is packaged as a compressed tar file.

The name of the Opswise Universal Agent for SOA 5.2.0 package file has the following format:

```
sb-soa-5.2.0.0-aix-5.3.tar.z
```

(The name assumes product maintenance level 5.2.0.0 for Opwise Universal Agent for SOA 5.2.0.)

To unpack and install Opwise Universal Agent for SOA, perform the following steps:

<b>Step 1</b>	Create a directory (or select an existing directory) in which to save the package file.
<b>Step 2</b>	Save the package file into that directory.
<b>Step 3</b>	<p>Uncompress and extract the installation files in the current working directory. The command to extract the files is:</p> <pre>zcat sb-soa-5.2.0.0-aix-5.3.tar.Z   tar xvf -</pre> <p>If your operating system does not support the <b>zcat</b> command, use the following command:</p> <pre>gunzip sb-soa-5.2.0.0-aix-5.3.tar.Z</pre> <p>The output of the <b>gunzip</b> command provides the following <b>tar</b> file: <b>tar -xvf sb-soa-5.2.0.0-aix-5.3.tar</b></p>
<b>Step 4</b>	<p>After the extraction is complete, run the installation script, <b>upsinst</b>, which executes the <b>installp</b> command:</p> <pre>./upsinst</pre> <p>An installation log is written to file <b>install.log</b> in the current directory. <b>upsinst</b> automatically restarts the Universal Broker daemon, <b>ubrokerd</b>, at the end of the install.</p>
<b>Step 5</b>	From the <b>license file</b> that was sent to you by Stonebranch, Inc., add the license information to the following file: <b>/etc/universal/uacs.conf</b>
<b>Step 6</b>	<p>Recycle Universal Broker using the following commands (cd to <b>/opt/universal/ubroker</b>)</p> <p>First:</p> <pre>./ubrokerd stop</pre> <p>Then:</p> <pre>./ubrokerd start</pre>

**Step 7** Use **Universal Query** (cd to `/opt/universal/bin`) to validate that the Universal Application Container Server component of Universal Command Agent for SOA 5.2.0 is running:

**uquery -host localhost** (or the name of your server)

The output should have the following format:

```
Component ID.....: 1360109684
Component Name.....: uac (Server)
Component Description....: Universal Application Container Server
Component Version.....: 5.2.0 Level 0 Release Build 101
Component Type.....: uac
Component Process ID.....: 23331000
Component Start Time.....: 18:14:42
Component Start Date.....: 02/05/15
Component Command ID.....: uac
Component State.....: REGISTERED
Component MGR UID.....:
Component MGR Work ID.....:
Component MGR Host Name...:
Component MGR IP Address..:
Component MGR Port.....:
Component Comm State.....: ESTABLISHED
Component Comm State Time.: 18:14:44
Component Comm State Date.: 02/05/15
Component MGR Restartable.: NO
Component Comment.....:
```

## 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.

## 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:

```
//XXXXXXXX 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
-mqueueename 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.
```

## 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  
-mqqueuemanagername MyQueueManager  
-mqqueueename 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.