Universal Command Agent for SOA 5.2.0

XD Connector

Getting Started with Universal Command Agent for SOA - XD Connector 5.2.0

Objective

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

- Installing Opswise Universal Agent for SOA 5.2.0, which is comprised of:
  - Universal Command Agent for SOA
  - Universal Event Monitor for SOA
- Validating the WebSphere environment.
- Running the XD Connector on Universal Command Agent for SOA.

Installation Requirements

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

- Opswise Universal Agent 5.2.0.0 or later (32-bit package): installed, licensed, and running.

Installation

Note
These instructions describe installation of the Opswise Universal Agent for SOA 5.2.0 for the Linux x86 package.

Opswise Universal Agent for SOA 5.2.0 is packaged as an RPM file (extension .rpm). It is installed using the Linux **rpm** command.

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

`sb-soa-5.2.0.0-linux-2.4-i386.tar.Z`

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

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

Step 1	Create a directory (or select an existing directory) in which to save the package file.
Step 2	Save the package file into that directory.

Step 3	Uncompress and extract the installation files in the current working directory. The command to extract the files is: `zcat sb-soa-5.2.0.0-linux-2.4-i386.tar.Z | tar xvf -` If your operating system does not support the **zcat** command, use the following command: **`gunzip sb-soa-5.2.0.0-linux-2.4-i386.tar.Z`** The output of the gunzip command provides the following **tar** file: **`tar –xvf sb-soa-5.2.0.0-linux-2.4-i386.tar`**
Step 4	After the extraction is complete, run the installation script, **upsinst**, which executes the **rpm** command: `./upsinst` An installation log is written to file **install.log** in the current directory. **upsinst** automatically restarts the Universal Broker daemon, **ubrokerd**, at the end of the install.
Step 5	From the license file that was sent to you by Stonebranch, Inc., add the license information to the **`/etc/universal/uacs.conf`** file.

Step 6	Recycle Universal Broker using the following commands (cd to **`/opt/universal/ubroker`**):
	
	First:
	
	`./ubrokerd stop`
	
	Then:
	
	`./ubrokerd start`

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.....: 02/05/13
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.: 02/05/13
Component MGR Restartable.: NO
Component Comment........:
```

Validating the WebSphere XD Environment

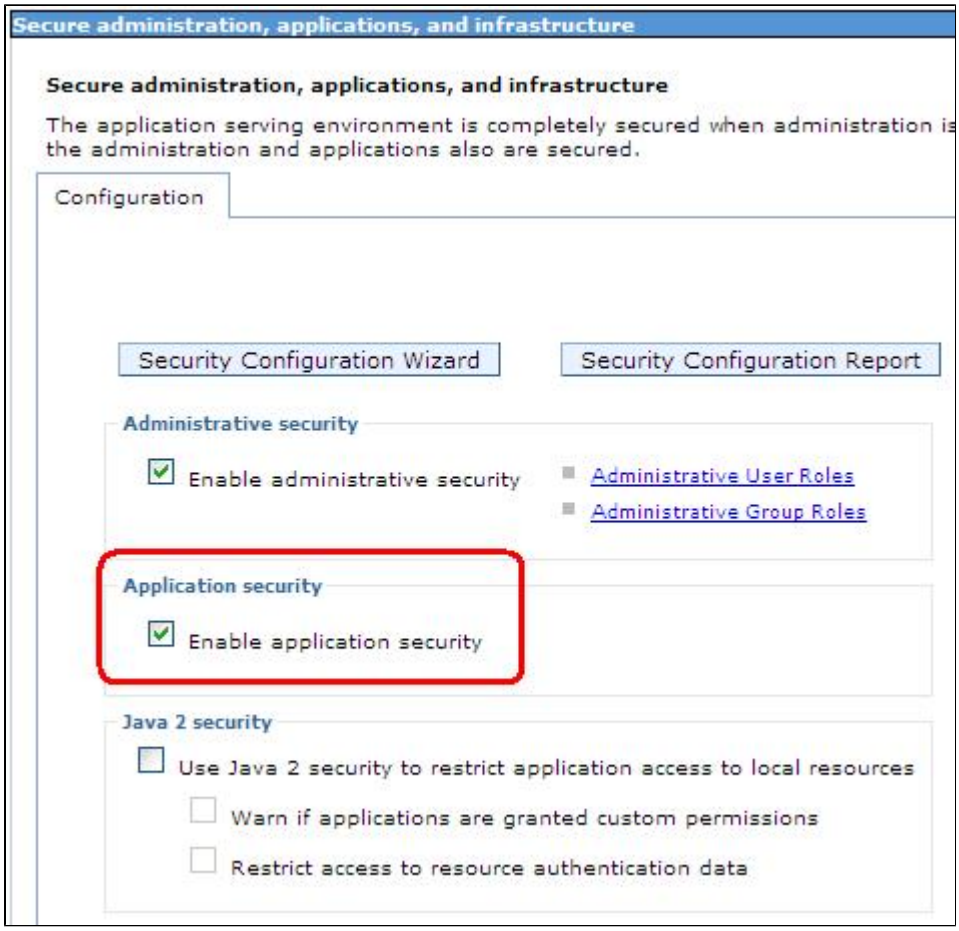
To ensure trouble-free operation of Universal Command Agent for SOA: XD Connector 5.2.0, the WebSphere XD environment must be set up correctly.

The following steps should be made before using the Universal Command Agent for SOA: XD Connector 5.2.0.

Step 1 **Application Security**

After installing the XD environment application, security must be enabled (if it is not on by default).

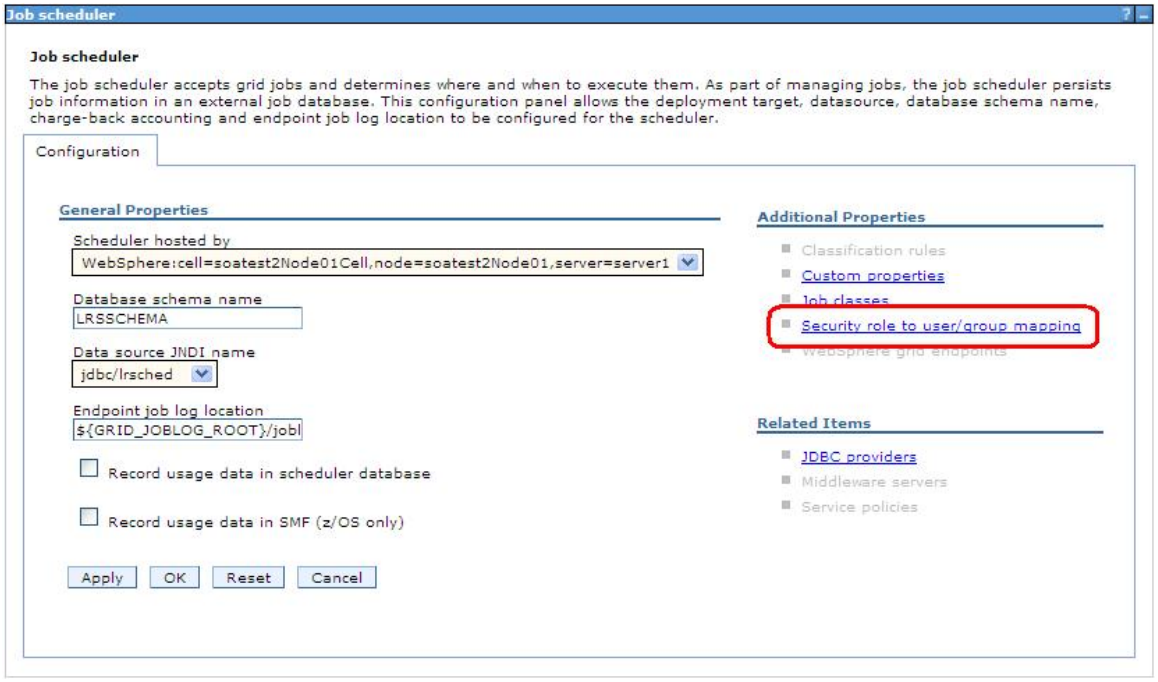
1. Select **Security** / **Secure administration, applications, and infrastructure** in the Integrated Solutions Console menu.
2. Check the **Enable application security** checkbox in the Application Security group.



Step 2 **Map Users to Job Scheduler**

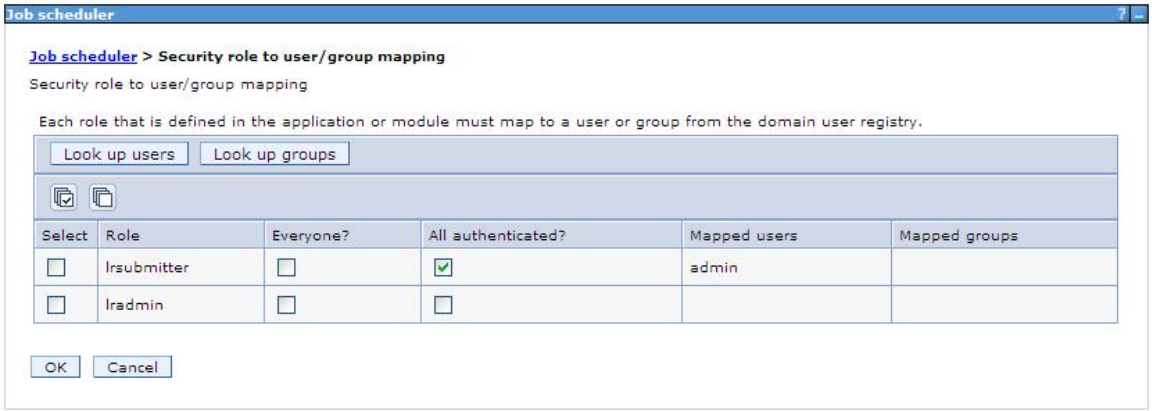
Users that have been set up in WebSphere must be mapped to a user in the job scheduler before they can run operations in the job scheduler.

1. Select **System administration** / **Job scheduler** in the Integrated Solutions Console menu.
2. Click **Security role to user/group mapping** in the Addition Properties group to display the **Security role to user/group mapping** page.



Assign WebSphere users to **lrsubmitter** or **lradmin** (or other job scheduler users if you have created them). Make sure the **All authenticated?** checkbox is checked.

The following screen illustrates the results of mapping the WebSphere user admin to the **lrsubmitter** job scheduler user.

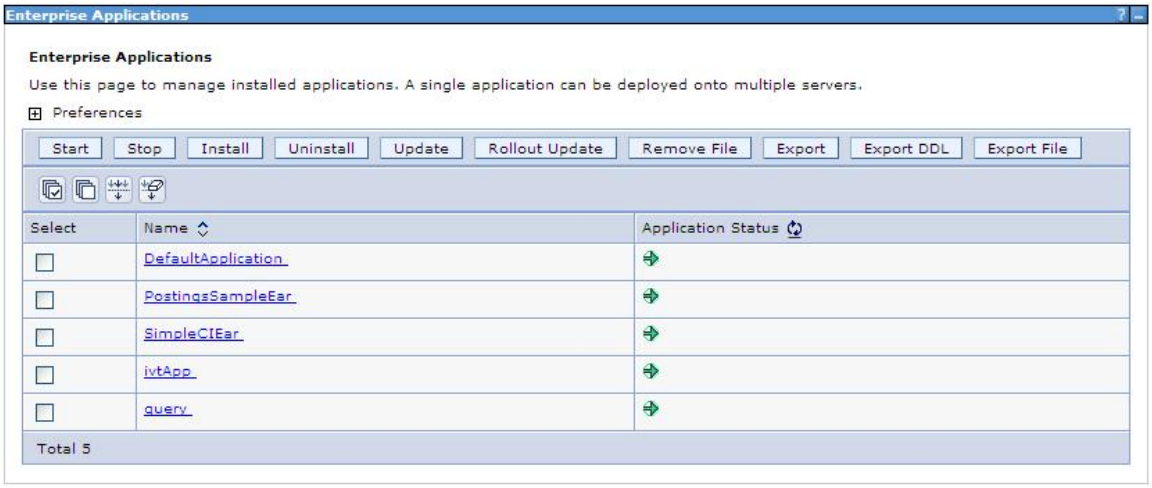


Step 3 **Install the Job Scheduler Sample Applications**

You must install the SimpleCIEar and PostingsSampleEar applications if you want to validate the WebSphere XD installation and the Universal Command Agent for SOA: XD Connector operation.

Go to **Applications** / **Install New Applications** in the Integrated Solutions Console menu. Keep in mind that the PostingsSampleEar application requires the database to be set up before you install the application.

The following screen illustrates the two applications after they are installed.



Step 4 **Run Sample Applications using `lrcmd.sh` Utility**

Now that you have successfully installed the sample applications, you can test that your WebSphere XD environment is set up correctly by running the applications using the `lrcmd.sh` utility.

1. The `lrcmd.sh` utility is located in **`/opt/IBM/WebSphere/AppServer/bin`** or where your local WebSphere installation path points to the bin directory.
2. The command syntax for submitting a job using `lrcmd.sh` is:

```
./lrcmd.sh –cmd=submit –
xJCL=/opt/IBM/WebSphere/AppServer/longRuning/postingSampleXJCL.xml –
host=localhost –port=9080 –userid=yourusername –password=yourpassword
```

> **Note**
> For the SimpleCIEar application, the xJCL is named SimpleCIxJCL.xml and host can be the name of any server on which WebSphere is installed.

The response to the submit command is:

```
CWLRB4960I: Fri Mar 28 15:20:43 EDT 2010 : com.ibm.ws.batch.wsbatch : Job [Posti
ngsSampleEar:18] is submitted.
```

3. When you have submitted the job, you can check its status by using the Job Management Console at https://yourservername:9443/jmc/console.jsp

   and running the **status** command.

   The syntax for using the **status** command is:
   `./lrcmd.sh –cmd=status –jobid=PostingsSampleEar:1 –host=localhost –port=9080 –userid=yourusername –password=yourpassword`

   The response to the status command is shown below:

```
CWLRB5160I: Fri Mar 28 15:24:46 EDT 2010 : Job [PostingsSampleEar:18] execution
has ended.
```

Running a Universal Command Agent for SOA Job on z/OS Connecting to XD Connector

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

The following z/OS examples reference the PostingsSampleEar application.

Step 1 **Create the UMCD Manager JCL.**

This provides the UCMD Manager options, references to the XD Connector options, and the xJCL payload.

It has the following format:

```
//UACXD1A  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*************************************************************
//*PostingsSampleEar
//*UCMD is the proc that calls UC Manager
//*LOGON is the DD with userid and passwd (can use encrypted)
//*SCR is the script that contains the XD Connector information //* to connect to Websphere job sche
//*UNVIN provides the payload for the script in SCR
//*************************************************************
//*
//         JCLLIB ORDER=SBI.UNV.SUNVSAMP
//*
//UCMD     EXEC UCMDPRC
//LOGON    DD  DISP=SHR,DSN=SUPPORT.UAC.LOGON(YOURUSER)
//SCR      DD  DISP=SHR,DSN=SUPPORT.UAC.SCRIPTS(SCRXD1)
//UNVIN    DD  DISP=SHR,DSN=SUPPORT.UAC.SCRIPTS(PYLXD1)
//SYSIN    DD  *
-s scr -script_type service
-i yourserver -G yes -f logon
```

Step 2 **Create the XD Connector Command Options Data Set Member.**

This member contains the command options for the XD Connector that specify the required information to submit a job to the WebSpher environment.

It is referenced with the **SCR** ddname and has the following format:

```
# protocol is what protocol you are choosing(XDSOAP)
# mep is the type of message (Request)
# xdcmd is the xd command you want to perform (submit,restart)
# cmdid is a unique identifier to match to jobid(websphere jobid)
# serviceurl is the websphere url to the job scheduler
# serviceusername is the websphere user
# servicepassword is the websphere user password
# timeoutms is the timeout in milliseconds for the job
#*******************************************
-protocol XDSOAP
-mep Request
-xdcmd SUBMIT
-cmdid 10010
-serviceurl http://sup\-wasxd:9080/LongRunningJobSchedulerWebSvcRouter/services/JobScheduler
-serviceusername admin
-servicepassword admin
-timeoutms 120000000
```

Step 3 **Create the xJCL Payload Data Set Member**.

This member contains the xJCL that WebSphere XD will use to run the job specified in the xJCL and is read via STDIN.

The xJCL length depends on the job it describes. Verify that your data set member record length can accommodate the maximum line le xJCL.

The xJCL is an XML document. It is referenced using the UNVIN ddname, and can be referenced in your WebSphere XD install by goin **`/opt/IBM/WebSphere/AppServer/longRunning/postingSampleXJCL.xml`**

(Your path may be different, depending on how you configured your WebSphere XD install.)