

Getting Started with Universal Command Agent for SOA: XD Connector 3.2.0

Objective

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

- Installing Universal Command Agent 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:

- Universal Products 3.2.0.0 or later: installed, licensed, and running.

Installation

These instructions describe installation of the Universal Command Agent for SOA for Linux package.

Universal Command Agent for SOA 3.2.0 is packaged as an RPM file (extension .rpm). It is installed using the Linux `rpm` command.

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

ups-3.2.0.0-linux-2.4-i386.tar.Z

The following steps describe the unpacking and installation of Universal Command Agent for SOA 3.2.0:

1. Create a directory (or select an existing directory) in which to save the package file.
2. Save the package file into that directory.
3. Uncompress and extract the installation files in the current working directory. The command to extract the files is:

```
zcat ups-3.2.0.0-linux-2.4-i386.tar.Z | tar xvf -
```

If your operating system does not support the `zcat` command, use the following command:

```
gunzip ups-3.2.0.0-linux-2.4-i386.tar.Z
```

The output of the `gunzip` command provides the following `tar` file:

```
tar -xvf ups-3.2.0.0-linux-2.4-i386.tar
```

4. After the extraction is complete, run the installation script, **upsinst**, which executes the **installp** 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.

5. You can use **uquery** (cd to **/opt/universal/bin**) to validate that Universal Command Agent for SOA 3.2.0 is running.

```
uquery -i localhost (or the name of your server)
```

The output should have the following format:

```
Component ID.....: 1206554190
Component Name.....: uac (Server)
Component Description.....: Universal Application Container Server
Component Version.....: 3.2.0 Level 0 Development Build 284
Component Type.....: uac
Component Process ID.....: 27070
Component Start Time.....: 01:56:29 PM
Component Start Date.....: 03/26/2008
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.....: STARTING
Component Comm State Time.: 01:56:29 PM
Component Comm State Date.: 03/26/2008
Component MGR Restartable.: NO
Component Comment.....:
```

6. From the license file that was sent to you by Stonebranch, Inc., add the license information to the **/etc/universal/uacs.conf** file.
7. Recycle **ubroker** using the following commands (cd to **/opt/universal/ubroker**):
 - a. **./ubrokerd stop**
 - b. **./ubrokerd start**
8. Again, you can use **uquery** (cd to **/opt/universal/bin**) command to validate that Universal Command Agent for SOA 3.2.0 is running after updating the **uacs** configuration file.

```
uquery -i localhost (or the name of your server)
```

Validating the WebSphere XD Environment

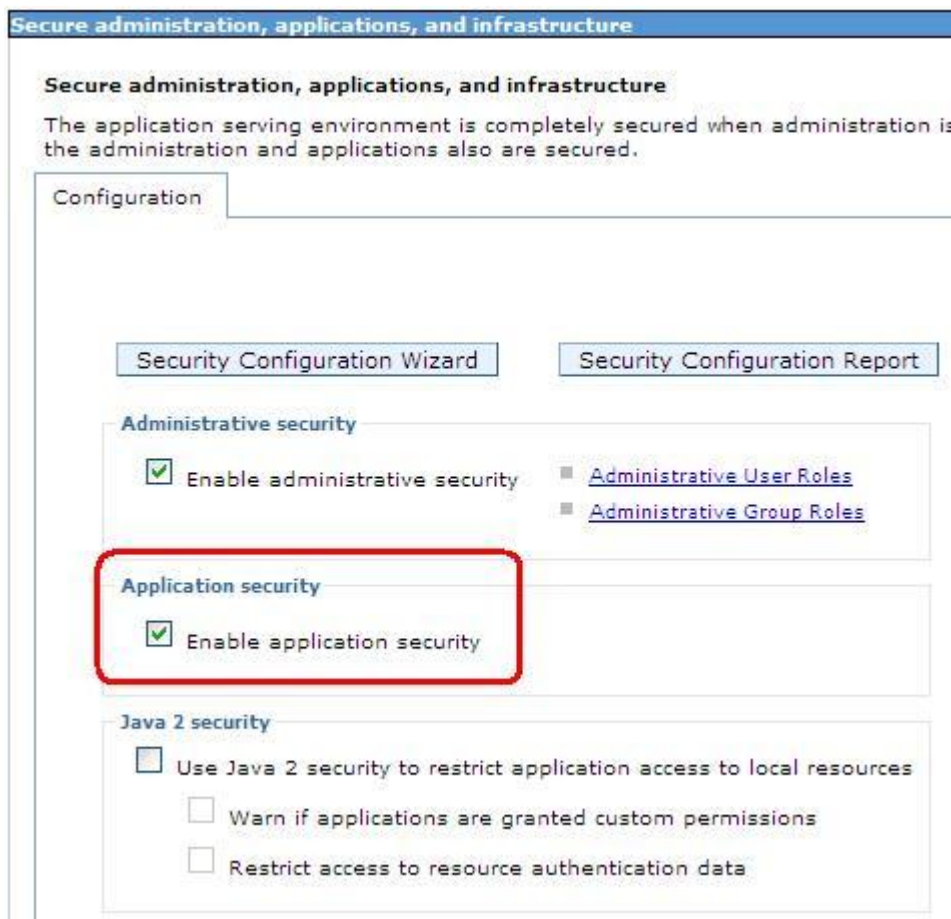
To ensure trouble-free operation of the Universal Command Agent for SOA: XD Connector 3.2.0, you must set up the WebSphere XD environment correctly.

Address the following areas before using the Universal Command Agent for SOA: XD Connector 3.2.0.

1. Application Security

After installing the XD environment application, you must enable security (if it is not on by default):

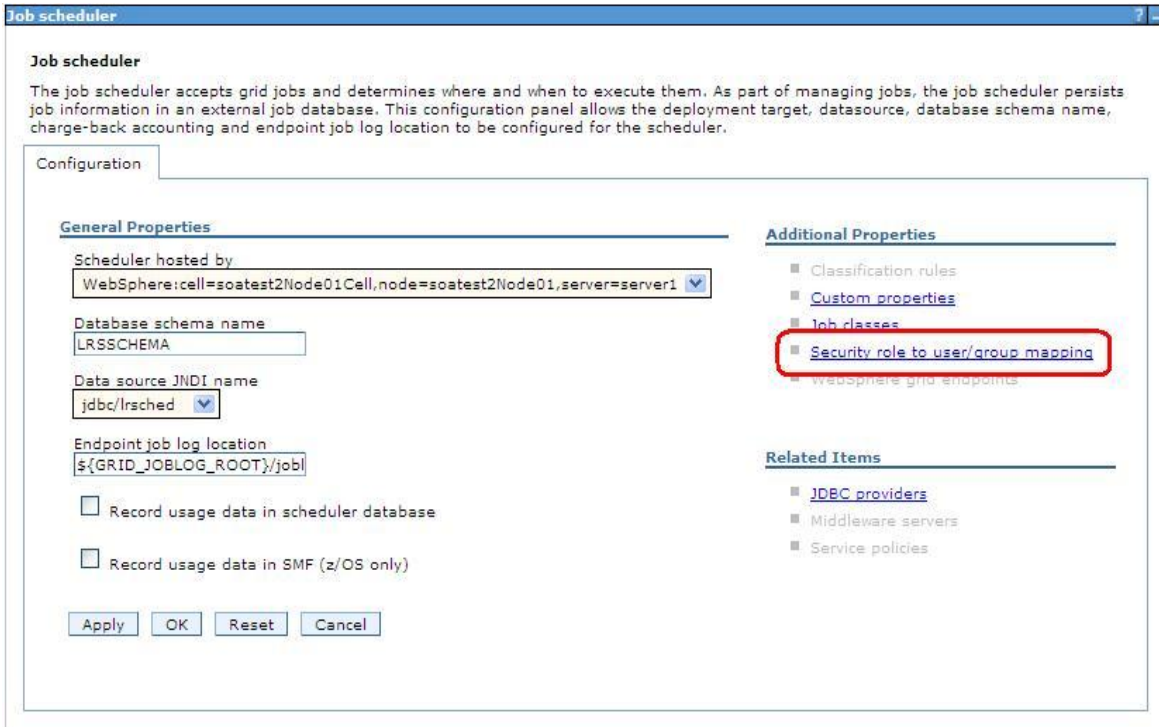
- a. On the **Integrated Solutions Console** menu, select **Security / Secure administration, applications, and infrastructure**.
- b. In the **Application Security** group, check the **Enable application security** checkbox.



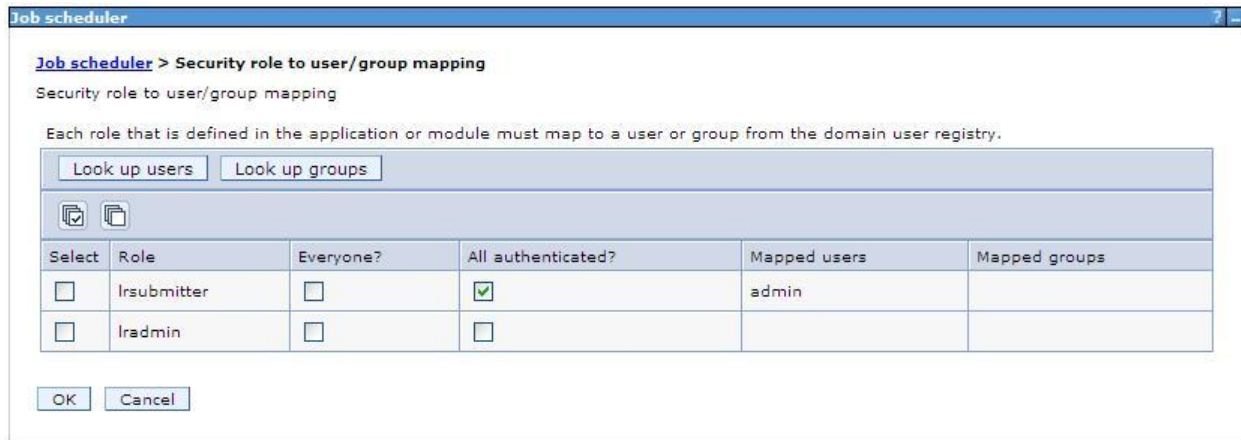
2. Map Users to Job Scheduler

You must map each user that has been set up in WebSphere to a user in the job scheduler before they can run operations in the job scheduler.

- a. On the **Integrated Solutions Console** menu, select **System administration / Job scheduler**.
- b. In the **Additional Properties** group, click on the **Security role to user/group mapping** link.



The **Security role to user/group mapping** page then displays.



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

This screen illustrates the results of mapping the WebSphere user admin to the lrsmitter job scheduler user.

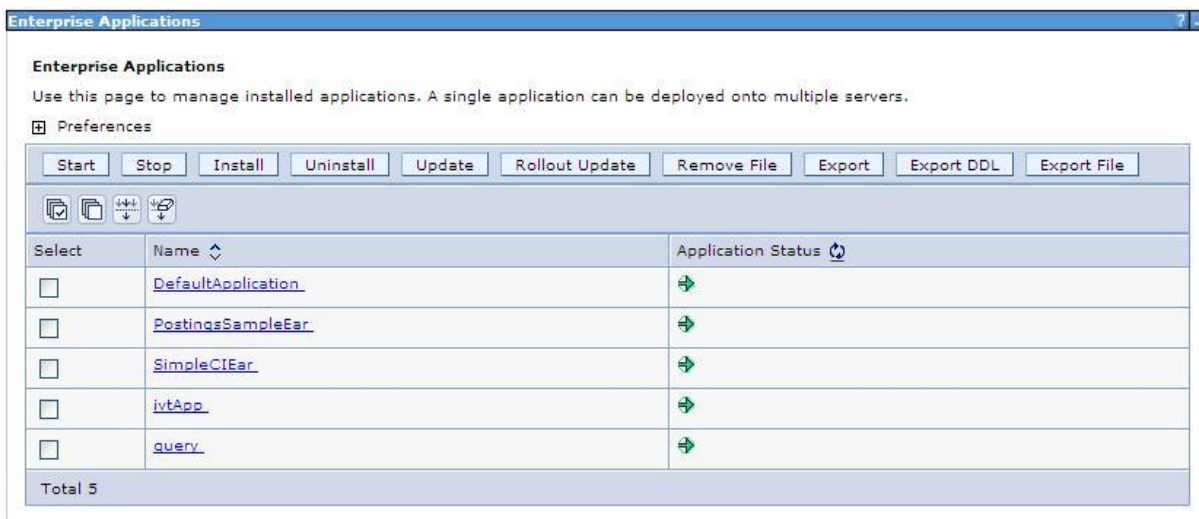
3. Install the Job Scheduler Sample Applications

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

On the **Integrated Solutions Console** menu, select **Applications / Install New Applications**.

Keep in mind that the PostingsSampleEar application requires that you set up the database before you install the application.

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



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

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

- a. The `lrcmd.sh` utility is located in `/opt/IBM/WebSphere/AppServer/bin` or wherever your local WebSphere installation path points to the `bin` directory.
- b. 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 shown below.

```
CWLRB4960I: Fri Mar 28 15:20:43 EDT 2008 : com.ibm.ws.batch.wsbatch : Job [Posti  
ngsSampleEar:18] is submitted.
```

- c. After you have submitted the job, you can check its status by using the Job Management Console through your favorite web browser at:

<https://yourservername:9443/jmc/console.jsp>

- d. After you have submitted the job, run 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 2008 : 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.

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 scheduler
//*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(PYLDX1)
//SYSIN     DD  *
-s scr -script_type service
-i yourserver -G yes -f logon
```

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 WebSphere XD 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
```

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 length in your xJCL.

The xJCL is an XML document. It is referenced using the UNVIN ddname, and can be referenced in your WebSphere XD install by going to:

/opt/IBM/WebSphere/AppServer/longRunning/postingSampleXJCL.xml

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