Universal Agent 6.7.x for z/OS

Quick Start Guide

Universal Agent 6.7.x for zOS Quick Start Guide

Objective

This Universal Agent 6.7.x for z/OS Quick Start Guide provides a short summary on how to install Universal Agent on a z/OS system and run a few simple batch jobs demonstrating Universal Agent on z/OS. For a complete description of the installation process, see Universal Agent 6.7.x Installation, Upgrade, and Applying Maintenance.

The following information is provided:

- System Requirements
- Downloading the Universal Agent 6.7.x for z/OS distribution file.
- Installing Universal Agent 6.7.x on z/OS.
- Licensing Universal Agent components.
- Encrypting passwords to be used in Universal Agent jobs.
- Executing a Universal Command job.
- Executing a Universal Data Mover job.
- Executing a Universal Broker query job.

System Requirements

- z/OS 2.2 or 2.3.
- SMP/E 3.5 or later.
- IBM Communication Server for z/OS 2.3 or 2.3.
- IBM Language Environment (LE) for z/OS 2.2 or 2.3.
- Windows workstation capable of establishing a TCP/IP network connection to the z/OS system.
- TSO user ID with an OMVS segment.
- About 1900 cylinders of DASD.
- Two available TCP/IP ports on z/OS.

Platform Requirements

Since platform requirements may change with new releases of a product, see Platform Support for Universal Controller 6.7.x and Universal Agent 6.7.x to make sure that your platform is supported before performing an installation.

Downloading the Universal Agent 6.7.x for z/OS Distribution File

The Universal Agent 6.7.x for z/OS product distribution file - `sb-6.7.x.x-zos.tar.Z` - is available for download from the Stonebranch Customer Portal.

A customer user name and password - provided by Stonebranch, Inc. - are required to access the Customer Portal.

Installing Universal Agent 6.7.x for z/OS

Step 1	Download one of the Universal Agent 6.7.x for z/OS product distribution files.
Step 2a	If one of the distribution files with the .tar.Z extension was downloaded, decompress it using any utility that can process files in Lempel-Ziv (that is, **compress**) format. Use **tar** to extract the files from the uncompressed file. To do both in a single statement, decompress the file to stdout and pipe the result to **tar** (for example, **zcat sb-v.r.m.l-zos.tar.Z | tar xvf -**).
Step 2b	If one of the distribution files with the .zip extension was downloaded, decompress it using any utility that can process files in **ZIP** (or, PKZIP) format. The .zip file can also be extracted using Windows native file extraction tools.
Step 3	Modify the UNVFTP.TXT script provided by the distribution file. This script contains statements to upload the installation files and sample JCL, which can be used to process the files after upload. Two versions of these files are included in the distribution file. The files in the main directory are EBCDIC-encoded files. ASCII-encoded versions of these files are in the `./ascii` subdirectory. See zOS Installation - Transferring Installation Files to zOS for complete details.
Step 4	Modify the JCL uploaded from UNVRECV.JCL and run the job to receive the TSO XMIT file(s). If the distribution file that contains the DFDSS archive is used, UNVRECV.JCL also contains a step to restore the contents of the archive. See Receive the Installation Datasets for complete details.
Step 5	Edit the INSTALL(#SETUP) member to meet local requirements as described in the #SETUP comment prolog. #SETUP creates customized installation JCL members in the INSTALL library. Execute the installation jobs as required for your SMP/E environment. See zOS Installation - SMPE Installation for complete details.
Step 6	Create started task user and group profiles for Universal Broker and Universal Enterprise Controller. See zOS Configuration - Started Tasks for complete details.
Step 7	Define the SUNVLOAD library to APF, Program Control, and LNKLST. See zOS Configuration - Load Library for complete details.
Step 8	Define SMF exits for Universal Command and Universal Automation Center Agent. See zOS Configuration - SMF Exits for complete details.
Step 9	Configure the UNIX System Services file systems used by the Universal Broker and Universal Enterprise Controller. See Universal Agent Database Configuration for complete details.
Step 10	Perform individual component customization, including JCL procedure edits and adding component licenses. See zOS Installation - Customization for complete details.
Step 11	Follow the optional steps described in z/OS USS Installation to complete the installation, if desired.

Licensing your Universal Agent 6.7.x for z/OS Components

After Universal Agent 6.7.x for z/OS has been installed, you must configure the following Universal Agent components with product licenses before they can be used:

- Universal Command Manager
- Universal Connector
- Universal Data Mover Manager

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

The format of a product license file name is: *<component name>_<customer name>_<operating system>_<schedule or solution>*.txt. For example: **Agent_Stonebranch_MVS_A1.txt**.

Universal Command Manager License File

For Universal Command Manager, **Agent** is used as the *<component name>* in the product license file name and as the name of the product in the file itself - see below.)

The following is a sample Universal Command Manager for z/OS product license file:

```
License_Product "AGENT"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

Universal Data Mover Manager License File

For Universal Data Mover Manager, **MFT** is used as the *<component name>* in the product license file name and as the name of the product in the file itself - see below.)

The following is a sample Universal Data Mover Manager for z/OS product license file:

```
License_Product "MFT"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

UNVCONF Product Library

In the **UNVCONF** product library:

* Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UCMCFG00**.
* Enter the Universal Connector license parameters into the Universal Connector configuration file, member **USPCFG00**.
* Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member **UDMCFG00**.

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	`STOP UBROKER`
Start Universal Broker	`START UBROKER[,UPARM='options']`

Encrypting User ID and Password for Job Execution

Universal Command and Universal Data Mover typically require user credentials to execute work on a remote host. The user credentials can be stored in an encrypted format for privacy reasons. Read access to the file should be restricted to authorized users even though the contents are encrypted. Universal Command and Universal Data Mover will read user credentials in the encrypted format.

The following JCL example executes the UENCRYPT utility that encrypts command options it reads from the UNVIN ddname and writes them to the UNVOUT ddname in the form of an encrypted command file. This example creates member DALLAS in the PDS/E **PROD.UNV.USERS**. The DALLAS member will contain the user ID and password options specified in the UNVIN ddname in a UENCRYPT format.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1    EXEC PGM=UENCRYPT
//STEPLIB  DD  DISP=SHR,DSN=#SHLQ.UNV.SUNVLOAD
//UNVIN    DD  *
 -USERID USR100 -PWD ABCDEF
/*
//UNVOUT   DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  DUMMY
```

Executing a Universal Command Job

The following JCL example executes the UNIX command "ls -l" on remote host DALLAS using the encrypted command file created above.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//         JCLLIB ORDER=#SHLQ.UNV.SUNVSAMP
//*
//STEP1    EXEC UCMDPRC
//USER     DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//SYSIN    DD  *
-cmd 'ls -l' -host dallas -encryptedfile user
/*
```

Executing a Universal Data Mover Job

Universal Data Mover execution is driven by an easy to use scripting language. The Universal Data Mover script controls the transfer sessions established with remote hosts and the file and transfer operations performed on the host's file systems.

The following JCL example executes Universal Data Mover. Universal Data Mover always reads the script from the UNVSCR ddname. In this example, the script establishes a session with remote host DALLAS and transfers the data set allocated to ddname DATA to file app.data on the remote host. The default transfer mode of binary is used for transferring the data.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1    EXEC UDMPRC
//DATA     DD  DSN=APP.DATA,DISP=SHR
//USER     DD  DSN=PROD.UNV.USERS(DALLAS),DISP=SHR
//UNVSCR   DD  *
set echo=yes halton=warn
open dest=dallas xfile=user
filesys local=dd
copy local=data dest=app.data
quit
/*
```

Running a System Query

Use the Universal Query utility to query the Universal Broker for Broker and active component information. Universal Query returns information about the Universal Broker, as specified by configuration options, as well as information regarding the components managed by Universal Broker.

The following JCL example executes the UQUERY utility. In this example, UQUERY will query the Broker running on host DALLAS.

```
//jobname  JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//         JCLLIB ORDER=#SHLQ.UNV.SUNVSAMP
//*
//STEP1     EXEC UQRYPRC
//SYSIN    DD  *
-host dallas
/*
```