Installation Guide
OAM REST API (OA64282)
V3R1 and above

Document Name:	OAM-RESTAPI-Installation-Guide-OA64282.docx
Document Owner:	Peter Sobik (pssobik@us.ibm.com)
Version:	V1.0

Version	Date	Change Description	Revision Tag
1.0	04/15/2025	Initial version	None

Overview and Requirements

The OAM REST API support consists of two parts: an OAM application deployed to a z/OS Liberty server and a native OAM z/OS program (referred to as the bridge program). These pieces work in conjunction to serve and authenticate S3 API requests and then forward the data to OAM's native OSREQ interface.

S3 methods and corresponding OSREQ functions supported are:

```
PutObject:          STORE
GetObject:          RETRIEVE
DeleteObject:       DELETE
HeadObject:         QUERY
GetObjectTagging:   QUERY
PutObjectTagging:   CHANGE
```

Requirements

z/OS V3R1 (and above)
z/OS Java 11 Semeru Runtime Certified Edition version 11.0.17.0 or higher
z/OS Liberty version 24.0.0.9 or higher

UNIX System Services (USS)
Integrated Cryptographic Service Facility (ICSF) with an AES Master Key defined as AES SHA256
IBM Crypto Express card
Resource Access Control Facility (RACF) or equivalent security product

Operator Commands

- To start the OAM bridge program – S OAMREST
- To stop the OAM bridge program – P OAMREST or F OAMREST,STOP
- To cancel the OAM bridge program – C OAMREST (used if a stop does not work)
- To display general information – F OAMREST,D,REST
- To display task information – F OAMREST,D,REST,TASK,[ALL | STORE | RETRIEVE | FAST]
- To cancel a task – F OAMREST,CANCEL,TASK,*taskname* (refer to command above for *taskname*)

Note: Refer to the User's Guide for details on the operator command support. "OAMREST" substituted as applicable for your environment.

1. OAM UNIX directory

This support requires that the following directory be created prior to installing OA64282.

`/usr/lpp/dfsms/oam`

Once the prerequisite OA65069 is installed, this can be done either by executing the DFPISMKD utility to run the DFPMKDIR EXEC from an SMP/E temporary directory, or by issuing the appropriate `mkdir` command to create the directory and then issuing `chmod` to set the directory permissions to 755. See the DFPISMKD utility for more information.

2. Install OAM APAR OA64282

Install the PTF for OA64282. The prerequisites are OA65068, OA65069 and OA66996.

After installation, an IPL is required.

This will create a new CBRWOLA AC(1) load module in LINKLIB and an oamrest.ear file in the `/usr/lpp/dfsms/oam` UNIX directory.

3. Program Properties Table

With APAR OA66996, the following Program Properties Table (PPT) entry is created in the SCHEDxx PARMLIB member:

```
PPT PGMNAME(CBRWOLA)   /* OAM RESTAPI PROGRAM PROPERTIES        */
         KEY(8)        /* PROTECT KEY ASSIGNED IS 8             */
         NOSWAP        /* PROGRAM IS NOT-SWAPPABLE              */
         SYST          /* PROGRAM IS A SYSTEM TASK              */
```

4. Create a z/OS Liberty server

Ensure that z/OS Liberty is installed. See Installing Liberty for additional information.

Create a z/OS Liberty server that will run the OAM RESTAPI application. It is recommended that this is the only application deployed on this server. See Creating a Liberty server manually for additional information.

5. Enable authorized services for z/OS Liberty

Grant the z/OS Liberty server authority to access the optimized local adapters service via the Liberty Angel process. This service is required by the OAM REST API. See Enabling z/OS authorized services on Liberty for z/OS for more information.

6. Install and enable required features in the z/OS Liberty server

Install the following features in the z/OS Liberty server. See Installing Liberty Repository assets for additional information.

```
jaxrs-2.1
zosLocalAdapters-1.0
jca-1.7
jndi-1.0
wasJmsClient-2.0 (required even if JMS messaging is not used)
```

Ensure the features are enabled by adding them to the z/OS Liberty server server.xml file. See Feature management for additional information.

```
<featureManager>
    <feature>jaxrs-2.1</feature>
    <feature>zosLocalAdapters-1.0</feature>
    <feature>jca-1.7</feature>
    <feature>jndi-1.0</feature>
    <feature>wasJmsClient-2.0</feature>
</featureManager>
```

7. Configure z/OS Liberty server properties

Add the following properties to the bootstrap.properties file in the z/OS liberty server directory. If this file does not exist, it must be created. See Specifying Liberty bootstrap properties for additional information.

`com.ibm.ws.zos.core.angelRequired=true`
Optional but recommended. This property will ensure that the Angel process configured in Enable authorized services for z/OS Liberty is available during server startup.

`com.ibm.oam.restapi.cda.group=`*`GROUPNAME`*
Required. This property specifies the Cloud Data Access (CDA) group name under which the S3 access keys are stored. See Add S3 credentials to Cloud Data Access (CDA) for additional information.

com.ibm.oam.restapi.region.*production=db2*
Optional. This property allows an alias to be specified for the region name on S3 requests.

By default, the S3 region name must contain the Db2 SSID of the OAM instance where this request will be directed. However, if this property is specified, when an S3 region name matches the final qualifier in this property, the application will substitute the region name for the value of this property for use as the Db2 SSID.

In the example above, a S3 request that specifies the region name of "production" will be directed to the OAM instance associated with Db2 SSID of "db2".

Note: There is no character limit on the alias name, but the Db2 SSID must be between 1 and 4 characters.

com.ibm.oam.restapi.bypass.secure=true
Optional and ***not recommended*.** This property will bypass a check which enforces a secure HTTPS connection via TLS/SSL. By default, the OAM RESTAPI web application will return HTTP 500 Internal Error if a non-secure connection is detected. Setting this property will disable this behavior and allow unsecure HTTP connections. This property is only intended to be used for installation verification purposes.

com.ibm.oam.restapi.osreq.codes=true
Optional. Allows the OSREQ return and reason code (success or failure) to be returned in the response header. The RETCODE2 value will also be returned even if just zero.

Java Message Service Properties

Refer to step 8 for additional information.

com.ibm.oam.restapi.jms.cf=*oam/jms*
Optional. Enables the Java Message Service (JMS) to be used for successful and/or failed requests.

com.ibm.oam.restapi.jms.*function*.queue=*oam/jmsqueue*
Optional. Defines the JMS message queue that OAM should use for the specified function **[store | retrieve | delete | query | change]**. The same queue or different queues can be used for the specified function. Messages can be sent for all functions or a subset of the functions.

com.ibm.oam.restapi.jms.*function*=[success | fail | all]
Optional. For the function specified **[store | retrieve | delete | query | change]**, indicates whether notifications will be sent for successful, failed, or all requests.

8. Setting up the Java Message Service (Optional)

To send event notifications, support for the Java Message Service is also being provided, refer to the following for setup:

- OAM Web Application will check for a new Java system bootstrap property com.ibm.oam.restapi.jms.cf to associate with the ConnectionFactory
    - bootstrap.properties example: com.ibm.oam.restapi.jms.cf=*oam/jms*
- **"*oam/jms*"** will correspond to the jmsQueueConnectionFactory defined in the server.xml.
    - Example server.xml definition:
    - <jmsQueueConnectionFactory jndiName=**"oam/jms**">
        - <properties.wasJms
        - nonPersistentMapping="ExpressNonPersistent"
        - persistentMapping="ReliablePersistent"/>
    - </jmsQueueConnectionFactory>
- OAM Web Application will support a message queue for each function (store, retrieve, delete, query and change). The messages will be sent to a queue defined by the client using the following Java system bootstrap property(s). The same queue can be specified for one or more of the functions or each function can be setup to use its own queue.
    - com.ibm.oam.restapi.jms.store.queue=*oam/jmsqueue*
    - com.ibm.oam.restapi.jms.retrieve.queue=*oam/jmsqueue*
    - com.ibm.oam.restapi.jms.delete.queue=*oam/jmsqueue*
    - com.ibm.oam.restapi.jms.query.queue=*oam/jmsqueue*
    - com.ibm.oam.restapi.jms.change.queue=*oam/jmsqueue*

  Example server.xml definition:

```
<jmsQueue jndiName="oam/jmsqueue">
  <properties.wasJms queueName="libertyQ"
   deliveryMode="Application"
   timeToLive="500000"
   priority="1"
   readAhead="AsConnection"/>
</jmsQueue>
```

- OAM Web Application will send event notifications for the following specified OSREQ functions (store, retrieve, delete, query and change). “Success” indicates that notifications will be sent for OSREQ return codes less than or equal to 4, “fail” indicates that notifications will be sent for OSREQ return codse greater than 4, and “all” indicates that notifications will be sent for all OSREQ requests.
    - com.ibm.oam.restapi.jms.store=[success|fail|all]
    - com.ibm.oam.restapi.jms.retrieve=[success|fail|all]
    - com.ibm.oam.restapi.jms.delete=[success|fail|all]
    - com.ibm.oam.restapi.jms.query=[success|fail|all]
    - com.ibm.oam.restapi.jms.change=[success|fail|all]

- Messages will be MapMessage objects with String key:value pairs for the following: function (store, retrieve, delete, query and change), collection name, object name, size and DB2 SSID, the request ID, and the OSREQ return and reason code.

  Example MapMessage:

```
size       : 659
requestid  : 5WSM1AD1EHD09Y0W
function   : store
osreq-rsn  : 00000000
collection : group01
osreq-rc   : 00000000
osreq-rc2  : 00000000
object     : group01.s001.s001
db2ssid    : dbb1
```

9. Enable the z/OS Liberty server to use optimized local adapters

Perform the following steps while referring to the corresponding notes for each step below: Enabling the Liberty server environment to use optimized local adapters

1. Follow steps as documented, no modifications required.
2. Follow steps as documented. After the optimized local adapters load module library has been created, it **must** be added to an authorized program facility (APF) list.
3. Follow steps as documented with the following modifications:
    a. No modifications.
    b. The three part WOLA group name can be chosen by the user. These three names will need to be specified during setup step Modify and run the CBRWPROC sample.

```
<zosLocalAdapters wolaGroup="NAME1"
                  wolaName2="NAME2"
                  wolaName3="NAME3"/>
```

4. Follow steps as documented. The jndiName **must** be defined as “oam/ola”. The RegisterName can be chosen by the user and must be unique for each Liberty server running on a z/OS LPAR. It will also need to be specified during setup step Modify and run the CBRWPROC sample.

   The connection manager must be defined to increase the maximum number of WOLA connections.

```
<connectionFactory jndiName="oam/ola"
                   connectionManagerRef="ConMgr1" >
  <properties.ola RegisterName="REGNAME"/>
</connectionFactory>

<connectionManager id="ConMgr1" maxPoolSize="2000" />
```

5. Follow steps as documented, no modifications required.
6. Follow steps as documented. Activate the CBIND class in System Authorization Facility (SAF) and create corresponding profiles to prevent unauthorized users or programs from accessing the WOLA group name.

10. Secure the z/OS Liberty server

Ensure that the z/OS Liberty server has been secured. Best practices on server hardening can be found at the CIS IBM WebSphere Liberty Benchmark (free account required). Clients and users are wholly responsible for determining the appropriate level of security required for their Liberty installation.

Complete a review of the Level 1 profile items in the following sections to determine if the item needs to be applied to this installation. It is also recommended to review all other sections to determine if any other items are applicable.

Section 1 - Install and Setup
Section 3 – Application Deployment
Section 4.2 – Secure Transport
Section 6 – Web Services
Section 9 – z/OS
Section 10 – Miscellaneous

Note: The OAM web application requires a secure connection via TLS/SSL or it will return HTTP 500. However, unsecured connections are supported but not recommended if the appropriate server property is configured. See Configure z/OS Liberty server properties for additional information.

11. Deploy the OAM REST API Application

Deploy the OAM REST API application to the z/OS Liberty server.

The enterprise application package file can be found at:

```
/usr/lpp/dfsms/oam/oamrest.ear
```

The default contextRoot of this application is:

```
<contextRoot>/</contextRoot>
```

See Deploying Applications in Liberty and Deploying a web application to Liberty for additional information.

12. Configure Cloud Data Access (CDA)

The Cloud Data Access (CDA) component of DFSMS is used to securely store the S3 credentials used by the OAM REST API. Perform the following steps to configure CDA to be used by the bridge program:

1. Create a home directory for the bridge program with appropriate permissions.

```
mkdir -m 770 bridge_home_name
```

2. Create subdirectories for Cloud Data Access configuration files with appropriate permissions.

```
mkdir -m 770 /bridge_home_name/gdk
mkdir -m 770 /bridge_home_name/gdk/providers
```

3. Create RACF Bridge Program User and Group IDs

RACF (or equivalent) user and group IDs for the bridge program must be created and configured. Below are example RACF commands that can be used to accomplish these steps:

a. Create a group specifically for the bridge program with a unique GID.

```
ADDGROUP bridgeg OMVS(GID(gid))
```

b. Create a user ID for the bridge program with a unique UID and assign it to the group above. A setting of **2000** per bridge program is recommend for **FILEPROCMAX** and **PROCUSERMAX**. Specify the home directory created in Step 1 as it is required to store CDA configuration files.

```
ADDUSER bridge DFLTGRP(bridgeg) OWNER(bridgeg) NAME('OAM Bridge
Program') NOPASSWORD OMVS(UID(uid) FILEPROCMAX(fff) PROCUSERMAX(ppp)
HOME(bridge_home_name))
```

c. Associate the bridge program started task with the bridge program group.

```
RDEFINE STARTED OAMREST*.OAMREST* STDATA(USER(=MEMBER)
GROUP(bridgeg))
```

Note: OAMREST is the default bridge program task name. It can be changed in Modify and run the CBRWPROC sample.

d. Refresh the RACF (or equivalent) profile.

```
SETROPTS RACLIST(STARTED) REFRESH
```

4. Create the RACF OAM administrator group

A RACF group for OAM administrators should be created with a unique GID.

Note: If an OAM administrator group already exists from a prior OAM/CDA setup, it may optionally be reused. If this is not desired behavior, create a new admin group to be used exclusively for OAM REST API administration.

```
ADDGROUP oamadmin OMVS(GID(gid))
```

Add appropriate users to this group. This includes:

- A security (or storage) administrator who will be responsible for adding OAM REST API S3 credentials via the CDA ISPF panels. See Add S3 credentials to Cloud Data Access (CDA) for more details.

5. Copy Cloud Data Access Configuration Files.

   a. Copy the provided sample JSON files to the OAM home directory: /usr/lpp/dfsms/gdk/samples/gdkconfig.json should be copied to: /*bridge_home_name*/gdk/config.json

   b. /usr/lpp/dfsms/gdk/samples/gdkkeyf.json should be copied to /*bridge_home_name*/gdk/gdkkeyf.json

c. Create a CDA provider definition file with appropriate permissions (770).

   The recommendation is to copy /usr/lpp/dfsms/gdk/samples/providers/BASIC.json to a new name under /*bridge_home_name*/gdk/providers

   The file must contain a valid CDA JSON cloud provider definition. The values contained within this file are not checked or utilized, however it still must be formatted correctly.

   The name of this copied file will represent the group under which all OAM REST API S3 credentials will be stored in Add S3 credentials to Cloud Data Access (CDA). This name should match the property specified in Configure z/OS Liberty server properties. **The group name must consist of all uppercase letters.**

   For example, if the specified server property was:

   `com.ibm.oam.restapi.cda.group=`*`GROUPNAME`*

   The corresponding CDA provider definition file would be:

   /bridge_home_name`/gdk/providers/GROUPNAME.json`

All file names are case sensitive.

6. Secure the bridge program home directory

Configure the OAM home directory with a Unix owner and group which allows only the bridge user ID and OAM administrator group access to the directory and files.

a. Change the bridge program home directory group to the OAM administrator group.

   `chgrp -R` *`oamadmin /bridge_home_name`*

b. Change the owner of the bridge program home directory to the bridge user ID.

   `chown -R` *`bridge /bridge_home_name`*

7. Configure the CSFKEYS general resource class

The CSFKEYS RACF (or equivalent) general resource class must be configured to allow CDA to utilize encryption services.

The CSFKEYS general resource class must be active and RACLISTed.

Define a profile for CSFKEYS resources beginning with GDK.** with a universal access (UACC) of NONE along with ICSF(SYMPACFWRAP(YES) SYMCPACFRET(YES)).

The bridge user ID must have READ access to a new CSFKEYS profile for resources beginning with GDK.*bridge*.** along with ICSF(SYMPACFWRAP(YES)SYMCPACFRET(YES)).

The security administrator or person who will be entering the cloud provider keys must have UPDATE access to the new CSFKEYS profile for resources beginning with GDK.*bridge.***

13. Add S3 credentials to Cloud Data Access (CDA)

Prior to this step, ensure the following:

That SYS1.DFQPLIB is part of the ISPPLIB concatenation or that the following members located in SYS1.DFQPLIB are added to an ISPPLIB library:

GDKAPPOP
GDKAUTHK
GDKAUTHL
GDKAUTHP
GDKMAINP
GDKOBJAC
GDKOBJAL
GDKCREDL (updated with OA65068)
GDKRESTC (new with OA65068)
GDKRESTK (new with OA65068)

A RACF (or equivalent) profile should be created to ensure only authorized users have access to these members.

Add one or more sets of S3 credentials to Cloud Data Access.

Issue the following TSO command:

EX 'SYS1.SAXREXEC(GDKRESTC)'

Select the provider or group name (previously configured in step 5.c), enter the “Bridge ID”and select option ‘O’ to continue to the next pane (GDKRESTK).

The ‘Bridge ID’ field should contain the RACF user ID under which the bridge program will run (previously configured in step 3).

Panel Contents

```
.  Menu  Options .
              z/OS Cloud Data Access OAM REST API Credentials Utility
 Option  ===> O
     L Display Credentials List
     O Open Credential Entry Panel

 Select Credentials Group Name
  _ 1.OAMREST




 Encryption Parameters
    Group Name . . .  OAMREST
    Bridge ID . . . . BRIDGEID

 Enter the Bridge ID, select the Credentials Group Name, and enter
   "O" on the Option to enter the Access Key and Secret Key.
```

On the following panel (GDKRESTK), enter appropriate values in the 'Authorization Parameters' and selection Option 'S' to encrypt and save the S3 credentials.

The **'RACF ID'** field should contain the RACF user id which will be associated with the S3 credentials. The RACF permissions of this user ID will be checked when an associated S3 access key is encountered. See Grant RACF Permissions for more information. This value **must** be upper case and a maximum of 8 characters.

The **'Access Key'** field should contain the S3 access key. This value **must** be unique to each set of credentials. Can contain maximum 142 case sensitive alphanumeric characters.

The **'Secret key'** field should contain the S3 secret key. Can contain maximum 303 case sensitive alphanumeric characters.

```
GDKRESTK Credential Entry Panel
  Menu  Options  Help
_____________________________________________________________________________
              z/OS Cloud Data Access OAM REST API Credentials Utility
 Option  ===> S
     S Save Credentials                       C Clear Secret Key Field
                                                 (for hidden input)
 Encryption Parameters
    Group Name . . .  OAMREST
    KeyLabel . . . .  GDK.BRIDGEID.OAMREST.Annnnn
    Keystore . . . .  /bridge_home/gdk/gdkkeyf.json

 Authorization Parameters
    RACF ID  . . USERID
    Access Key . AccessKey

    Secret Key . **********************************************



 Enter the RACF ID associated with the credentials along with the Access Key
 and Secret Access Key that will be used to access the OAM REST API.
```

14. Grant RACF Permissions

Ensure that the RACF FACILITY class is ACTIVE and that the RACF user id(s) specified in setup step Add S3 credentials to Cloud Data Access (CDA) have been granted read permission to the following RACF profile:

Class: `FACILITY`
Profile: `STGADMIN.CBR.RESTAPI`

If the profile does not exist, it must be created. The use of wildcards in the profile is permitted if a more specific profile has not been defined.

Note: Refer to the User's Guide for function specific profile checks. For example, a STORE request could check the profile name: `STGADMIN.CBR.RESTAPI.STORE`.

15. Modify and run the CBRWPROC sample

Modify and run the CBRWPROC job found in SAMPLIB. See the documentation in the sample job for additional information.

16. Start OAM REST API components

Ensure that the appropriate OAM address space has been started with object support enabled, then perform the following steps

1.) Start the Angel process.
2.) Start the z/OS Liberty server.
3.) Start the OAM bridge program.

The OAM REST API is now started ready to accept S3 requests.

Diagnostic Guidance

- Each S3 request will have a unique RequestID assigned as an identifier. The RequestID will be returned in the HTTP response header and body. Any corresponding errors messages will be written to `stderr` with this unique RequestID. Output can be found in /logs/messages.log (default location) in the Liberty server directory. This includes any errors encountered during the OSREQ invocation and will include the corresponding return and reason codes.

- Security audit records can be found in /logs/messages.log (default location) in the Liberty server directory. Each S3 request that is served by the application will get a corresponding audit record. The output message JSON formatted and line delimited:

```
{
    "time": "2023-03-29T20:42:10.582Z",
    "requestid": "7SFNSFM4Q1VTKRZN",
    "httpmethod": "GET",
    "authorized": true,
    "accesskey": "s3accesskey",
    "user": "userid",
    "bucket": "group01",
    "objectkey": "group01.s001.s001",
    "region": "production",
    "db2ssid": "db2"
}
```

- To enable detailed logging of the OAM Liberty application, add the following line to the server.xml file:

  ```
  <logging traceSpecification="com.ibm.oam.*=finest"/>
  ```

  Output can be found in /logs/trace.log (default location) in the Liberty server directory.

- Other (higher level) logging levels (info, audit, warning, and error) can also be enabled, refer to the Usage Guide for additional information. By default, the log level is set to “info” which will also capture “audit”, “warning” and “error” log entries.

- If an 0C4-10 abend occurs in the OAMREST (CBRWOLA) address space (CSECT CBRWORPT), we believe that the cause of this abend is in the Liberty code and surfaces when the following GETMAIN/FREEMAIN traps are enabled in your DIAGxx PARMLIB member:

  IARCP64INITFREE
  IARCP64INITGET
  IARCP64TRAILER
  IARST64INITFREE
  IARST64INITGET
  IARST64TRAILER

  If you see these abends, these traps may have to be temporarily disabled until a fix is found. Refer to the following DIAGxx reference link:
  https://www.ibm.com/docs/en/zos/3.1.0?topic=trace-statements-parameters-diagxx

- Review your Db2 configuration to determine the CTHREADS/MAXDBAT and/or IDBACK values.

  While REST requests are being sent to OAM (through OSREQ), you'll want to monitor your Db2 configuration to determine if the numbers specified for CTHREADS, MAXDBAT and IDBACK need to be raised (or lowered). With OAM's REST support, the OAM Bridge Program (per task) conntects to Db2, and then for each PUT (STORE), GET (RETRIEVE) and DELETE request, a DB2 thread is gotten and then released as the request is processed. At most the OAM Bridge Program could consume 200 Db2 threads for PUTS (STORES), 200 Db2 threads for GETS (RETRIEVES) and 200 Db2 threads for deletes if there is enough work and everything is concurrently being processed. However, as a REST request is processed, a Db2 thread will be gotten and will be released, freeing up Db2 threads for other tasks. You'll want to take this into account when determining the numbers to specify for CTHREADS, MAXDBAT and IDBACK. The values specified are a balance between performance and Db2 resource consumption. If a task in the OAM Bridge Program cannot get a Db2 thread, the task will wait until a Db2 thread is made available.

- If you are seeing warning message "**SRVE8115W: WARNING: Cannot set status. Response already committed",** the following setting in the server.xml file may alleviate these messages:

  ```
  <webContainer com.ibm.ws.webcontainer.invokeflushafterservice="false" />
  ```

  This issue was fixed in OpenLiberty but it was not back to the z/OS code base. See the following OpenLiberty GitHub issue for additional information: https://github.com/OpenLiberty/open-liberty/issues/10988

- As tests are running, the *F OAMREST,D,REST,TASK* command can be issued to monitor the active OAMREST tasks that are processing work.

- To validate that the RACF user id and group for the bridge program have been correctly associated with the started task, check the following message in the system log after starting the bridge program:

  ```
  IEF695I START OAMREST  WITH JOBNAME OAMREST  IS ASSIGNED TO USER
  bridge, GROUP bridgeg
  ```

- To validate the bridge program RACF user id UNIX setup, issue the following TSO command:

  LISTUSER *bridge* OMVS

  Example Output:

```
USER=bridge
DEFAULT-GROUP=bridgeg
PHRASEDATE=N/A
ATTRIBUTES=PROTECTED
REVOKE DATE=NONE   RESUME DATE=NONE
LAST-ACCESS=20.272/11:07:04
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
…
OMVS INFORMATION
----------------
UID= 0000012345
HOME= bridge_home_name
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAX= 00002000
PROCUSERMAX= 00002000
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

- If the Angel process was started with a NAME parameter, ensure the com.ibm.ws.zos.core.angelName Liberty property is specified with the appropriate named Angel.

Appendix

WebSphere Application Server for z/OS Liberty

CIS IBM WebSphere Liberty Benchmark