

# OA64874 Publication Updates

10/23/2023

Andrew Wilt (anwilt@us.ibm.com)

## 1 Enhancement Overview:

A customer requested an enhancement to the GDKUTIL program, specifically the ability to store metadata about the cloud object along with the object during an UPLOAD command.

<https://bigblue.aha.io/ideas/ideas/ZOS-I-3616>

<https://ibm-z-hardware-and-operating-systems.ideas.ibm.com/ideas/ZOS-I-3616>

The DFSMSdftp Cloud Object Utility (GDKUTIL) will be enhanced to allow specification of metadata to be associated with the Cloud Object on an UPLOAD command.

### 1.1 Enhancement Details:

#### 1.1.1 Metadata for Uploaded Objects

The GDKUTIL program is enhanced to accept a new keyword, METADATA(<dd\_name>), on the SYSIN for an UPLOAD command that specifies the DD name containing the key:value pairs that should be sent to the Cloud Object server for association with the object.

The provider file for the cloud object storage server will also need to be updated with new support for a different type of JSON object in the requestParameters object, named METAHEADER.

requestParameters new field for meta data description

```
{
  "mechanism": "METAHEADER",
  "descriptor": "x-amz-meta-"
},
```

The METAHEADER header instruction will contain a “descriptor” that contains the prefix to be used for each piece of metadata.

#### 1.1.2 Displaying Metadata for an Object

The sponsor user, mentioned that it would be great if z/OS could display the metadata tags for an object. Currently they use the free/trial Cyberduck program to view the metadata for an object. Since the metadata is stored as tags with the object, a HEADOBJECT request will receive those tags in the headers that are returned for the request.

The LIST function of the GDKUTIL program is enhanced to display the header information for a specific object when a full object name is specified.

Additionally, the LIST functionality of GDKUTIL is enhanced to display a list of bucket names when only the / character is specified as the ‘bucket’ to LIST.

## 1.2 z/OS MVS Programming: Callable Services for High Level Languages

SA23-1377-60

Chapter 25. Cloud Data Access Files is updated in the Provider file section:

Under **requestParameters**, a new section is added as follows:

### **METAHEADER**

Describes how to construct a metadata header.

The “descriptor” key value contains the header prefix for metadata headers that are built as part of the metadata optional parameter processing.

Chapter 30. Cloud Data Access Callable Services is updated in the following sections:

### **GDKWRITE – Write an object to cloud storage**

The table describing the acceptable Optional Parameters is updated to add a new row as follows:

metadata	Character	<p>Null terminated string of comma-separated key:value pairs. These key value pairs will be turned into HTTP headers according to the METAHEADER entry in the requestParameters object in the GETOBJECT operation. The key portion will be folded to lowercase and appended to the descriptor value from the METAHEADER entry in order to create a provider appropriate metadata header tag. The value specified must be less than 2048 characters in length and may be:</p> <ul style="list-style-type: none"><li>• A constant value, which will be added as-is. Contents are not examined for sensitive data.</li><li>• A System symbol specified with the ampersand character. i.e. &amp;SYSNAME</li><li>• A JCL symbol specified with the ampersand character (&amp;)</li><li>• A CDA Symbol specified with the ampersand character (&amp;). See table below for list of supported CDA symbols.</li></ul> <p>If a symbol does not resolve to a value, it is ignored and not included in the metadata attached to the cloud object.</p> <p>The WRITEOBJECT, or WRITELARGEOBJECT operation in the provider file must have a mechanism:HEADER object with a META type in the requestParameters that describes the prefix specific to that Cloud Storage provider. The sample provider files include this. If the provider file does not have the META type object, the request will fail.</p>
----------	-----------	--

		An HTTP header will be created for each key:value pair in the comma separated string.
--	--	---

Table nn: List of CDA symbols and their meaning. The value is resolved at execution for the uploaded data set or UNIX file if applicable.

CDA symbol	Meaning
GDK_BLKSIZE	Block size value from the z/OS data set
GDK_LRECL	LRECL value for the z/OS data set
GDK_RECFCM	Record Format value for the z/OS data set <ul style="list-style-type: none"> <li>• F – Fixed records</li> <li>• FB – Fixed Blocked</li> <li>• FBA – Fixed Blocked ASA print-control characters</li> <li>• FBS – Fixed Blocked Standard</li> <li>• FS – Fixed Standard</li> <li>• V – Variable</li> <li>• VB – Variable Blocked</li> <li>• VBA – Variable Blocked with ASA print-control Characters</li> <li>• VBS – Variable Blocked Spanned records</li> <li>• U – Unknown</li> </ul>
GDK_DSORG	Data Set organization <ul style="list-style-type: none"> <li>• PS – Physical Sequential</li> <li>• PDS – Partitioned Data Set</li> <li>• LIBRARY – Partitioned Data Set Extended</li> </ul>
GDK_EXPDATE	Expiration Date of the cataloged z/OS data set
GDK_CREDATE	Creation Date of the cataloged z/OS data set or UNIX file
GDK_REFDATE	Last Reference Date of the cataloged z/OS data set or last backup date of a z/OS UNIX file.
GDK_ISDATASET	True if z/OS data set. False if z/OS UNIX file
GDK_DATACLAS	Data Class associated with data set
GDK_MGMTCLAS	SMS Management Class associated with data set
GDK_STORCLAS	SMS Storage Class associated with data set
GDK_ONAME	Original Data Set or File name
GDK_JOBNAME	JCL Jobname that put this object into the Cloud
GDK_STEPNAME	JCL Step name that put this object into the Cloud
GDK_CTIME	Time the z/OS UNIX file metadata changed
GDK_MTIME	Time the z/OS UNIX file data changed

### GDKLIST – List cloud objects

The **bucketName** parameter first paragraph description is updated to the following:

Specifies the address of a pointer to a null-terminated string for the list request. The string may have the following attributes:

- / - A single forward slash character. This indicates that this is a request to list all available bucket names. The LISTBUCKETS operation must exist in the provider file.
- /<bucket\_name>/ - The remote bucket name that should be listed. The name passed must be null-terminated. The bucket/container name is folded to lowercase. The LISTOBJECT operation in the provider file will be used to perform the operation and format the result into the GDK\_OBJECTDESCRIPTION\_TYPE entries.
- /<bucket\_name>/<object-name> - The full object name that should be listed. The name passed must be null-terminated. The bucket name is folded to lowercase. The HEADOBJECT operation in the provider file must exist and will be used to fill the GDK\_OBJECTDESCRIPTION\_TYPE entry. Only information about the single object will be returned. The optional parameter, Get-Header-Buffer and Get-Header-Buffer-Size, may be used to pass a buffer that will be filled with all of the HTTP headers from the response.

The name is not checked for valid URL characters. For more information, see RFC 3986 ([tools.ietf.org/html/rfc3986](http://tools.ietf.org/html/rfc3986)).

Additionally, the Optional Parameters table is updated to add an additional two rows as follows:

Get-Header-Buffer	Address	Pointer to storage area to place the headers from the HTTP Response.
Get-Header-Buffer-Size	Integer	Size of the provided header buffer. Must be provided if Get-Header-Buffer is specified.

### 1.3 z/OS DFSMSdfp: Utilities

SC23-6864-60

Chapter 2. GDKUTIL (Cloud Object Utility) Program is updated in the SYSIN statement section to modify the row for List as follows:

List	<p>Requests a list of appropriate information from the cloud server according to the specified object name specified on the OBJNAME DD:</p> <ul style="list-style-type: none"> <li>• / - Requests a list of the accessible buckets from the cloud object server. The LISTBUCKETS operation must exist in the provider file.</li> <li>• /&lt;bucket_name&gt;/ - Requests a list of the objects in the named bucket. The LISTOBJECT operation must exist in the provider file.</li> <li>• /&lt;bucket_name&gt;/&lt;object_name&gt; - Requests a listing of the metadata and other information from the time of the request. The HEADOBJECT operation must exist in the provider file.</li> </ul>
------	--

Chapter 2. GDKUTIL (Cloud Object Utility) Program is updated in the SYSIN statement section to add a new row under the Keywords as follows:

METADATA	<p>Optional. Specifies a DD name that contains descriptions of the metadata to be saved with the cloud object during an UPLOAD command. The contents of the metadata DD can specify key:value pairs, separated by commas, where the key is folded to lowercase and used in a provider appropriate HTTP header. The key cannot contain blank or space characters.</p> <p>The value specified must be less than 2048 characters in length and may be:</p> <ul style="list-style-type: none"> <li>• A constant value, which will be added as-is. Contents are not examined for sensitive data.</li> <li>• A System symbol specified with the ampersand character. i.e. &amp;SYSNAME</li> <li>• A JCL symbol specified with the ampersand character (&amp;)</li> <li>• A CDA Symbol specified with the ampersand character (&amp;). See table below for list of supported CDA symbols.</li> </ul> <p>If a symbol does not resolve to a value, it is ignored and not included in the metadata attached to the cloud object.</p> <p>The WRITEOBJECT, or WRITELARGEOBJECT operation in the provider file must have a mechanism:METAHEADER object in the requestParameters that describes the prefix specific to that Cloud Storage provider. The sample provider files include this. If the provider file does not have the METAHEADER object, the request will fail.</p>
----------	--

Table nn: List of CDA symbols and their meaning. The value is resolved at execution for the uploaded data set or UNIX file if applicable.

CDA symbol	Meaning
GDK_BLKSIZE	Block size value from the z/OS data set
GDK_LRECL	LRECL value for the z/OS data set
GDK_RECFM	<p>Record Format value for the z/OS data set</p> <ul style="list-style-type: none"> <li>• F – Fixed records</li> <li>• FB – Fixed Blocked</li> <li>• FBA – Fixed Blocked ASA print-control characters</li> <li>• FBS – Fixed Blocked Standard</li> <li>• FS – Fixed Standard</li> <li>• V – Variable</li> <li>• VB – Variable Blocked</li> <li>• VBA – Variable Blocked with ASA print-control Characters</li> <li>• VBS – Variable Blocked Spanned records</li> <li>• U – Unknown</li> </ul>
GDK_DSORG	Data Set organization

	<ul style="list-style-type: none"> <li>• PS – Physical Sequential</li> <li>• PDS – Partitioned Data Set</li> <li>• LIBRARY – Partitioned Data Set Extended</li> </ul>
GDK_EXPDATE	Expiration Date of the cataloged z/OS data set
GDK_CREDATE	Creation Date of the cataloged z/OS data set or UNIX file
GDK_REFDATE	Last Reference Date of the cataloged z/OS data set or last backup date of a z/OS UNIX file.
GDK_ISDATASET	True if z/OS data set. False if z/OS UNIX file
GDK_DATACLAS	Data Class associated with data set
GDK_MGMTCLAS	SMS Management Class associated with data set
GDK_STORCLAS	SMS Storage Class associated with data set
GDK_ONAME	Original Data Set or File name
GDK_JOBNAME	JCL Jobname that put this object into the Cloud
GDK_STEPNAME	JCL Step name that put this object into the Cloud
GDK_CTIME	Time the z/OS UNIX file metadata changed
GDK_MTIME	Time the z/OS UNIX file data changed

The GDKUTIL Examples section is expanded with an additional example as follows:

### Example 3: Send a GDG version to a Cloud Object with Metadata tags

In this example, a sequential data set with EBCDIC data is sent to cloud object storage with conversion to UTF-8. Metadata tags are associated with the cloud object according to the names and symbols specified.

```
//UPCLOUD EXEC PGM=GDKUTIL,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
  UPLOAD PROVIDER(IBMCO) CONVERT
    METADD(DSMETA)
/*
//OBJNAME DD *
/cloudbucket01/previous-report.txt
/*
//LOCNAME DD *
  APPLICAT.SMS.REPORT(-2)
/*
//DSMETA DD *
  blksize:&GDK_BLKSIZE,lrecl:&GDK_LRECL,dsorg:&GDK_DSORG,
  sysname:&SYSNAME,jobname:&GDK_JOBNAME,stepname:&GDK_STEPNAME,
  recfm:&GDK_RECFM,expiredate:&GDK_EXPDATE,createdate:&GDK_CREDATE,
  dataset:&GDK_ISDATASET,dclass:&GDK_DATACLAS,mclass:&GDK_MGMTCLAS,
  sclass:&GDK_STORCLAS,origname:&GDK_ONAME,
  referenceDate:&GDK_REFDATE,sample:MySampleJCL
/*
```

## 1.4 z/OS MVS System Messages: Vol 5 (EDG-GRU)

SA38-0672-60

Chapter 10. GDK Messages is updated to add new messages as follows:

### 1.4.1 GDKU0017W

GDKU0017W INVALID SYNTAX FOR METADATA. <tag> MUST BE IN KEY:VALUE FORMAT

#### Explanation

The metadata tag list needs to be comma separated entries. Each tag must be in the format of a key name followed by a colon character (:) followed by the value requested.

#### System Action

This tag entry is skipped. Processing continues with other tags. Return code is 4.

#### Operator Response

None

#### System Programmer Action

Examine the contents of the DD specified by the METADATA keyword. <tag> identifies the tag in error.

#### Source

DFSMSdfp CDA

### 1.4.2 GDKU0018E

GDKU0018E ERROR USING fldata() FOR <ddname> ERRNO: <error\_desc>

#### Explanation

A system service error occurred while using the fldata() function for the named DD statement <ddname>. <error\_desc> describes the error returned from the system service.

#### System Action

Processing stops. Overall return code is 8.

#### Operator Response

None

#### System Programmer Action

Examine the named DD along with the error description in <error\_desc>. Resolve the issue and re-run the command.

#### Source

DFSMSdfp CDA

### **1.4.3 GDKU0019E**

GDKU0019E UNABLE TO OPEN <ddname> ERRNO: <error\_desc>

#### **Explanation**

The named DD statement was attempted to be opened to gather the requested attribute information, but was unable to. <error\_desc> gives more information about the error received.

#### **System Action**

Processing stops. Overall return code is 8.

#### **Operator Response**

None

#### **System Programmer Action**

Use the error description in <error\_desc> to identify the issue before re-running the command.

#### **Source**

DFSMSdfp CDA

### **1.4.4 GDKU0020I**

GDKU0020I UNKNOWN CDA METADATA SYMBOL <symbol>

#### **Explanation**

A CDA symbol (starting with &GDK\_ ) was specified, but is unknown to CDA. The symbol in error is displayed by <symbol>

#### **System Action**

Processing continues without that symbol being resolved.

#### **Operator Response**

None

#### **System Programmer Action**

Examine the contents of Metadata DD for the identified symbol name, and correct it for the supported CDA symbols.

#### **Source**

DFSMSdfp CDA



### 1.4.5 GDKU0021I

GDKU0021I METADATA DD <ddname> IS EMPTY.

#### Explanation

The specified DD <ddname> contained no data.

#### System Action

Processing continues without any metadata added to cloud object.

#### Operator Response

None

#### System Programmer Action

Consider whether metadata is desired to be added with the requested cloud object.

#### Source

DFSMSdfp CDA

### 1.4.6 GDKU0022I

GDKU0022I UNABLE TO OBTAIN INFORMATION ABOUT {DATA SET <dsname> {ON VOLSER: <volser>  
OBTAIN RC=<rc>}} | {UNIX FILE <filename> ERRNO: <error\_info>}

#### Explanation

The GDKUTIL utility was unable to find information about the named data set <dsname>. If the message contains the VOLSER, then an error occurred with the OBTAIN macro. The return code is additionally displayed. Information about the error can be seen in the Return Codes from OBTAIN (Reading by Data Set Name) in the z/OS DFSMSdfp Advanced Services manual.

The GDKUTIL utility was unable to find information about the named UNIX file <filename>. <error\_info> indicates the error information returned as a result of the stat64() call.

#### System Action

Processing stops with return code 8.

#### Operator Response

None

#### System Programmer Action

Check the data set name to ensure it is a cataloged data set. Check the volume <volser> to ensure that the cataloged data set exists on the volume. Check to ensure the UNIX file exists.

#### Source

DFSMSdfp CDA

### **1.4.7 GDKU0024W**

GDKU0024W {KEY <key>}{VALUE <value>} TOO LONG

#### **Explanation**

The metadata DD held a key or value that exceeds the allowed length. The maximum key name length is 255 characters. The maximum value length is 2048 characters.

#### **System Action**

Processing continues without the requested key included in the metadata for the object.

#### **Operator Response**

None

#### **System Programmer Action**

Check the metadata DD for the named key. Ensure it does not exceed allowed lengths.

#### **Source**

DFSMSdfp CDA