

Publication Updates for OA65451

The OA65451 APAR introduces a new API called GDKLISTL allowing callers to receive information about objects that are larger than 4GB in size. The GDKLIST API only supports returning an size that is 4GiB in size. If an object has a size larger than 4*1024*1024*1024, the size value returned for that object when calling GDKLIST will be incorrect.

What follows are the publication updates documenting this new API.

MVS Programming: Callable Services for High Level Languages.

SA23-1377-60

Chapter 29. Syntax, linkage, and programming considerations is updated to add a new row to the table listing the API Entry Offsets from DFVCDAVT.

96 (60)	GDKLISTL	NO
---------	----------	----

Chapter 30. Cloud Data Access callable services is updated to add the following section.

GDKLISTL — List cloud objects

The GDKLISTL API is used to list objects in a bucket for the specified cloudProvider.

Description

The GDKLISTL API is used to retrieve a list of objects in a bucket for a cloud provider. When called with a cloud provider and a bucket name and a buffer, an HTTP request is sent to the cloud object storage server to retrieve the information according to the entry in the LISTOBJECT entry in the supportedOperations array in the cloud provider json file, is sent and information about the cloud objects returned. It is different from the GDKLIST API in that it supports reporting object sizes that are larger than 4*1024*1024*1024 (4GiB).

The GDKLISTL API processing can additionally be modified through the optionalParmStruct.

Syntax

```
gdklistl (returnCodeAddr,  
         cloudProvider,  
         bucketName,  
         buffer,  
         bufferLen,  
         optionalParmStructPtr);
```

Parameters

The parameters are explained as follows:

retCodeAddr

Specifies the address of a 4-byte field that the API will place the return code into.

cloudProvider

Specifies the address of a pointer to a name of the cloud provider that this request should contact. The name must be null-terminated. When retrieving the cloudProvider definition corresponding to the cloudProvider specified on the API call, the user's RACF ID, or UserID from the optionalParms, is used to examine the associated OMVS segment to retrieve the home directory. That home directory is examined for a gdk/providers/sub-directory. If it exists and the <cloudProvider>.json file exists, it will be used as the template to communicate with the cloud provider. If the gdk/providers/ directory does not exist or the <cloudProvider>.json file is not found there, then the CDA system default directory of /usr/lpp/dfsms/gdk/providers/ will be used.

The application may request a list of all supported providers by using the **GETPROVIDERS** operation.

bucketName

- / - 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.

When interacting with a provider using the AWS4 authentication model, some characters are encoded as follows:

Character	Encoded version
/	%2F
"encode": "special" in provider file	
!	%21
*	%2A
(%28

buffer

Specifies the address of a pointer to a buffer where returned data should be placed. The buffer is formatted as described by the GDK_LISTLARGEOBJECTS_TYPE structure as detailed:

GDK_LISTLARGEOBJECTS _TYPE

GDK_LISTLARGEOBJECTS _TYPE				
Offset	Len	Type	Name	Description
0		STRUCTURE	GDK_LISTLARGEOBJECTS _TYPE	Structure of buffer containing LIST results.
0	4	FIXED	GDK_NumberOfObjects	Number of GDK_OBJECTDESCRIPTION_TYPE entries in the buffer.
4	2	FIXED	gdklist_ver	Version of gdklist
6	2	FIXED	_rsrvd	Reserved for future use
8	*	ARRAY	GDK_ObjectDescArray	Array of GDK_LARGEOBJECTDESCRIPTION_TYPE entries

Table 136. GDK_LARGEOBJECTDESCRIPTION_TYPE

GDK_LARGEOBJECTDESCRIPTION_TYPE

Offset	Len	Type	Name	Description
0		STRUCTURE	GDK_LARGEOBJECTDESCRIPTION_TYPE	Structure of One LIST result.
0	4	ADDRESS	GDK_objectName	Pointer to a null-terminated object name.
4	4	FIXED	createdTime	Number of seconds since January 1, 1970 when Object was created or -1 if unset.
8	4	FIXED	modifiedTime	Number of seconds since January 1, 1970 when Object was modified.
12	4	FIXED	isDirectory	1 = pseudo directory
16	8	FIXED	objectSize	Size of objects in bytes.
24	12	FIXED	list_reserved	reserved for future use

The GDKLISTL API will process the response body according to the LISTOBJECT operation's contentType and specified schema detailing the structure of the response. If contentType is application/json, it will be processed as a json document. If contentType is application/xml, it will be processed as an XML document.

Additionally, the provider definition may specify "contentType": text/plain in the responseResults section of the LISTOBJECT operation. This will cause the buffer to be filled

with the raw data returned from the Cloud Provider. Additionally, in this case, the return code will be set to GDK_LIST_RAW (300) to indicate that the buffer is filled with raw data and is not mapped by the structures above.

bufferLen

Specifies the address of a 4-byte signed field that contains the length of the passed buffer.

optionalParmStructPtr

Specifies an optional method for a user of this API to provide customized processing not provided by default by the CDA API. The API will specify a pointer to a structure as mapped by the data structure **GDK_OPTIONAL_PARMS_TYPE**. This data structure will contain one or more customized overrides or additions. In general, all string values, including keys and values, must be null-terminated when being passed to the APIs.

Optional Parameters

NAME	TYPE	Description
UserID	8-byte char	RACF UserID used to retrieve cloud security credentials for the DELETE request.
Get-HWTH-Code	Address	Specifies the address of a 4-byte Integer field where the resulting z/OS Client Web Enablement Toolkit return code should be stored. For more information, see "HWTHRQST — Send a request to an HTTP server" .
Get-HWTH-Diag	Address	Specifies the address of an HWTH_DIAGAREA_TYPE area where the resulting z/OS Client Web Enablement Toolkit diagnostic information should be stored. For more information, see "HWTHRQST — Send a request to an HTTP server" .
Set-Header-Buffer	Character	String of custom headers that should be included on the HTTP DELETE request. header is newline (\n) separated.
Use-Config-File	Character	false means that the config.json file should not be read for default configuration values. Any other values mean that the config.json should be used.
log-level	Character	The logging level can be set as desired. This value will override any default, or value in the config.json file. Logging messages are written to stdout. The levels in order of low to high severity are listed: <ul style="list-style-type: none"> • DEBUG means all logging messages are written to. • INFO means only INFO and higher severity logging messages are written. • NOTICE means only NOTICE and higher severity logging messages are written. • WARNING means only WARNING and higher severity logging messages are written. • ERROR means only ERROR logging messages are written. • NONE means no messages are written regardless of severity.

web-toolkit-logging	Character	<p>false means that additional logging messages from the z/OS Client Web Enablement Toolkit processing should not be written.</p> <p>true means that additional logging messages from the z/OS Client Web Enablement Toolkit processing should be written.</p> <p>This value will override any default or value in the config.json file. Logging messages are written to stdout.</p>
---------------------	-----------	--

Related services

“GDKGET — Retrieve a cloud object”

“GDKWRITE — Write an object to cloud storage”

“GDKDEL — Delete an object from cloud storage”

“GDKLIST — Get list of cloud objects”

Usage notes

- When the API is invoked, if UserID wasn't provided in the optionalParms, the current user's RACF ID is used to retrieve the applicable cloudProvider definitions, as well as retrieve the appropriate security credentials for the User/Cloud Provider/Resource combination from the gdkkeyf.json document.
- When retrieving the cloudProvider definition corresponding to the cloudProvider specified on the API call, the user's RACF ID, or UserID from the optionalParms, is used to examine the associated OMVS segment in order to retrieve the home directory. That home directory is examined for a gdk/providers/ sub-directory. If it exists and the <cloudProvider>.json file is found, it will be used as the template to communicate with the cloud provider. If the gdk/providers/ directory does not exist or the provider file is not found there, then the CDA System Default directory of /usr/lpp/dfsms/gdk/providers/ will be used.
- When retrieving the security credentials to use to communicate with the cloudProvider specified on the API call, the gdkkeyf.json file must first be located. The user's RACF ID, or UserID from the optionalParms, is used to find the home directory. That home directory is examined for a gdk/gdkkeyf.json document that can be accessed for READ. If not found, then the CDA API call fails.
- Once the keyfile has been found, it is parsed, looking for an entry associated with the user's RACF ID, or UserID from the optionalParms. The most specific bucket/container name (Resource) is searched for first. A bucket name is the first characters surrounded by a forward slash character (for example, /bucket_name/). If no entry is found, then the generic / Resource name is searched for.
- With the most appropriate entry found, the security credentials will be decrypted using ICSF. The decryption key is expected to be found using a keylabel of GDK.<cloudProvider>.<UserID/RACF ID>.<sequence_number>. If RACF protection of the ICSF key labels is being used, the invoker of the API must have authority to the key label.
- This API is supported on z/OS 2.5 and higher.

Restrictions

None.

Authorization

User.

Return and reason codes

The various return code constants are documented in the gdkic header file, found in SYS1.SIEAHDV.H.

Return code	Constant Name	Explanation
0	GDK_OK	Processing was successful.
100	GDK_UNABLE_TO_READ_KEYFILE	The gdkkeyf.json document was not found in the CDA System directory.
102	GDK_UNABLE_TO_PARSE_KEYFILE	When parsing the gdkkeyf.json document, it is not a valid JSON document.
104	GDK_CLOUD_PROVIDER_NOT_FOUND	When reading the gdkkeyf.json document, for the requested user, the specified cloudProvider was not found.
105	GDK_NO_RESOURCE_FOUND_IN_KEYFILE	For the specified UserId, and cloudProvider, no credentials entry for / was found.
110	GDK_PROVIDER_OPEN_FAILURE	The JSON document for the requested cloudProvider does not exist, or was found, but cannot be opened for READ.
111	GDK_PROVIDER_SPECIFICATION_INVALID	When parsing the JSON document, it was found to be invalid.
112	GDK_FEATURE_UNSUPPORTED	An unknown contentType was found in the parameterSet.
113	GDK_BUFFER_TOO_SMALL	When decrypting the security credentials, the specified buffer was too small to hold the decrypted credential.
114	GDK_AUTH_INIT_FAILURE	When using the AWS4 authentication model, the appropriate entry in the gdkkeyf.json document for the MVSUserID, cloud provider, and name (resource) is missing either the key, or the secretkey key-value pairs.
116	GDK_AUTH_APPLY_FAILURE	An error occurred while applying the authorization parameters.
118	GDK_USER_INFO_NOT_FOUND	When reading the gdkkeyf.json document, the requested user entry was not found.
123	GDK_CONN_NOT_HTTPS	HTTPS was not specified for httpMethod in the provider file.
799	GDK_UNEXPECTED_ERROR	An unexpected error occurred. Contact IBM Level 2, and provide the logging output.
800	GDK_CONNECTION_FAILED	Unable to connect to the host set in the cloud provider JSON definition.
801	GDK_TOOLKIT_FAILED	Error occurred in the z/OS Client Web Enablement Toolkit HWTHRQST call.

900	GDK_OBJECT_NOT_FOUND	HTTP status 404 was returned indicating the object was not found.
901	GDK_ACCESS_DENIED	HTTP status 403 was returned indicating the user is not authorized to access the resource in the desired manner.
902	GDK_OPERATION_NOT_SUPPORTED	When parsing the cloud provider JSON document, the WRITEOBJECT description was not found in the supportedOperations array.
903	GDK_RESPONSE_FORMAT_MISMATCH	The request was successful, but the data returned was in a format that did not match what is expected according to the contentType specified in the responseResults for the action in the provider file.
904	GDK_REQUEST_FAILED	A bad HTTP status (4xx or 5xx) was returned.