z/OS 3.1 IBM Education Assistant

Solution Name: z/OS data set and file rest service enhancements

Solution Element(s): z/OS data set and file REST interface

July 2023

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Objectives

Explain how z/OS data set and file rest service enhancements can help z/OS system programmer and application developer use z/OSMF REST API based data set or USS file operations in more scenario so that they can be more efficient.

Overview

- Who (Audience)
  - z/OS system programmer and application developer
- What (Solution)
  - Support three new headers for TSO procedure, account number and region size.
  - Support read/write data set and dataset members with names in different encodings.
  - Support nosetuid option when mounting UNIX file system.
  - Support create and remove a symbol link.
  - Support one new header to wrap the data when writing to a F format data set.
- Wow (Benefit / Value, Need Addressed)
  - Users can set the TSO procedure, account number and region size by new headers for each request.
  - Users can read/write data set and dataset members with names in different encodings so that their national special characters can be used in data set name.
  - Users can mount a UNIX file system with nosetuid option. Without the nosetuid option, a mount request requires a user have authority to the SUPERUSER.FILESYS.MOUNT resource which is usually reserved for just administrators or sys progs. With nosetuid, a mount request requires authorization to SUPERUSER.FILESYS.USERMOUNT which an average user is more likely to have.
  - Users can create and remove a symbol link.
  - Users can set a new header to wrap the data to avoid truncation error when writing to a F format data set and the original data number in one line is bigger than data set record length.

Usage & Invocation

z/OS data set and file rest service supports three new headers for TSO procedure, account number and region size.

- One new request header (X-IBM-Request-Proc=<string>) allows the request to set the TSO/E logon procedure. If you do not use this header, the RESTAPI_FILE PROC value specified in z/OSMF parmlib is used as a TSO/E logon procedure name. If you use this header but specify a string with length smaller than 1 or longer than 8, the request fails. Otherwise, it overrides the RESTAPI_FILE PROC value specified in z/OSMF parmlib.
- One new request header (X-IBM-Request-Acctnum=<string>) allows the request to set the TSO account number. If you do not use this header, the RESTAPI_FILE ACCT value specified in z/OSMF parmlib will be used as TSO/E logon procedure account number like before. If you use this header but specify a string with length smaller than 1 or longer than 40, the request will fail, otherwise it will override the RESTAPI_FILE ACCT value specified in z/OSMF parmlib.
- One new request header (X-IBM-Request-Region=<integer>) allows the request to set the TSO region size. The region size unit is K. If you do not use this header, the RESTAPI_FILE REGION value specified in z/OSMF parmlib is used as a TSO/E logon procedure region size. If you use this header but specify an integer with a size smaller than 1 or longer than 2096128, the request fails. Otherwise, it overrides the RESTAPI_FILE REGION value specified in z/OSMF parmlib. If you set the value of this header to be less than 65536, the system defaults to TSO region size to 65536K.

- Notes: These three header take effective only when creating a new TSO address space. If the request is reusing an existing TSO address space, they will be ignored.

Usage & Invocation

z/OS data set and file REST API can support data set and data set members with different encodings so that their national special characters can be used in data set name.

A new request header (X-IBM-Dsname-Encoding=<string>) will allow the request to set the encoding of the names of data set or member rather than IBM-1047.

One restriction is that the data set or member name character’s UTF-8 code can be converted to a valid IBM-1047 character.

Usage & Invocation

z/OS data set and file REST API users can set nosetuid option when mounting UNIX file system.

In the past, z/OS data set and file REST API services support mounting a z/OS UNIX file system on a specified directory by read only mode(rdonly) or read/write mode(rdwr). This enhancement supplies nosetuid mode. To use mount nosetuid option, users can set the mode field to "rdonly nosetuid“ or "rdwr nosetuid”.

Usage & Invocation

z/OS data set and file REST API users can create and remove a symbol link by z/OS Unix file utilities services.

Users can create a symbol link or external link by set the request value to "link" and set type value to "symbol" or "exteranl". Users also can remove a link by setting the request value to "unlink”.

For example, users can create a symbol link by below request body: {“request”:“**link**”, “from”:“/u/test.txt”,“type”:“symbol”}. If users want to remove the link, just need set the request key to unlink in JSON.

Usage & Invocation

z/OS data set and file REST API users can set wrap header to avoid truncation error when writing to a F format data set and the original data number in one line is bigger than data set record length.

The request header (X-IBM-Data-Type=<text;wrap=true>) allows the request to set the data wrap to avoid truncation error.

Currently, when writing text data to a F or FB dataset and dataset members, a truncation error occurs caused by the input text line length larger than the record length. When this header is set to TRUE, we will not report a truncation error but wrap the data into data set.

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level:
- No
- List any toleration/coexistence APARs/PTFs.
- N/A
- List anything that doesn't work the same anymore.
- N/A
- Upgrade involves only those actions required to make the new system behave as the old one did.
- N/A
- Coexistence applies to lower level systems which coexist (share resources) with latest z/OS systems.
- N/A

Installation & Configuration

- List anything that a client needs to be aware of during installation and include **examples** where appropriate - clients appreciate these:
  - Are any APARs or PTFs needed for enablement? No additional APARs are needed for V3.1.
  - What jobs need to be run? N/A
  - What hardware configuration is required? N/A
  - What PARMLIB statements or members are needed? N/A
  - Are any other system programmer procedures required? N/A
  - Are there any planning considerations? N/A
  - Are any special web deliverables needed? N/A
  - Does installation change any system defaults? N/A

Summary

- Two enhancements for z/OS data set and file rest service enhancements have been explained.

Appendix

- Please refer to z/OSMF Configuration Guide and z/OSMF Programming Guide for more details about z/OS data set and file rest service.