z/OS V1R13

Language environment:
LE/C BSAM >64K tracks (binary and text)

Overview (1 of 2)

- Problem statement / need addressed
  - Sequential data sets can only use a maximum of 59 volumes. There is a limit for the number of tracks a volume can contain. Once the limit has been reached, there is no room for the data set to grow.
  - In z/OS V1R7, DFSMSdfp provided support for large format sequential data sets, removing the size limit of 65,535 tracks/volume for QSAM, BSAM, and EXCP
  - C/C++ applications need to be able to exploit the functionality added by DFSMSdfp.
  - The basic forms of ftell() and fseek() can not handle the offsets that are possible when working with large format sequential data sets. These functions only support the reporting of and repositioning (directly or relatively) to offsets that are 2 GB – 1 or less.
- Solution
  - Provide XL C/C++ run-time library support for large format sequential data sets greater than 65,535 tracks/volume when opened for BSAM (seek) under binary and text I/O.
  - Allow large files versions of ftello() and fseeko() to work on large format sequential data sets.
- Benefit / value
  - The ability for an XL C/C++ application to grow sequential data sets beyond the 65,535 tracks/volume limit.

Usage and invocation

- Using LI177, you can:
  - Use XL C/C++ run-time library routines to process large format sequential data sets that are greater than 65535 tracks/volume using **binary** or **text I/O**.
  - Provides applications with the ability to read from, write to, and reposition (seek) within data sets that are greater than 65,535 tracks/volume using common OS I/O routines.
- The support is invoked by calling the fopen() function on a pre-existing large format sequential data set (DNSTYPE=LARGE was specified).
  - Note: Allocation of a new large format sequential data set can be accomplished by specifying the keyword DSNTYPE=LARGE on a JCL DD statement or using the dynamic allocation equivalent..
  - New macro __DSNT_LARGE added for use with dynalloc() function.
  - Support for the fopen() or freopen() equivalent of DSNTYPE=LARGE is **not** added by LI177
- Once the data set is opened, other OS I/O functions can be used to process the stream.
- Supported data set types:
  - The XL C/C++ run-time library OS I/O routines can be used on large format sequential data sets that are single or multivolume. Data sets can reside on SMS-managed or non-SMS managed storage devices. They can also be catalogued or uncatalogued.
- New/Changed external output:
  - An ABEND during OPEN/CLOSE/EOV processing is no longer generated when seek is used with large format sequential data sets greater than 65535 tracks under binary or text I/O.
  - Attempts to open a large format sequential data set for read (r,rb,rt) with **seek** under binary or text I/O while the data set is already open for write (w,wb,wt) with **noseek** will no longer fail.
- LI177 also adds the capability to use the large files versions of the ftello() and fseeko() functions on MVS data sets that are opened for any type of I/O (record, binary or text).
- Allows reporting of and repositioning, either directly or relatively, to offsets greater than 2GB – 1.
- In order to use the large files versions of these functions, define the following feature test macro in your application:

```
#define _LARGE_FILES 1
```

Interactions and dependencies

- Software dependencies
  - None
- Hardware dependencies
  - DASD capable of containing a large format sequential data set larger than 65,535 tracks on the volume.
- Exploiters
  - C/C++ applications that require I/O support for data sets larger than 65,535 tracks per volume

Migration and coexistence considerations

- None

Installation

- None

Session summary

- XL C/C++ run-time library routines support BSAM (seek) processing of large format sequential data sets greater than 65,535 tracks/volume under binary and text I/O.
- Large files version of ftello() and fseeko() support offsets greater than 2 GB – 1 for MVS data sets

Appendix - References

- XL C/C++ run-time library Reference (SA22-7821)
- XL C/C++ programming Guide (SC09-4765)