IBM Education Assistance for z/OS V2R2

Item: Long Symbols
Element/Component: BCP Supervisor

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Migration & Coexistence Considerations
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- Understand the new Long Symbols capability

Overview

- Problem Statement / Need Addressed
  - Symbol names are too short. Requiring values to be no longer than the name is too restrictive.
- Solution
  - Allow longer symbols with longer-still names (with restrictions)
- Benefit / Value
  - Improved usability

Usage & Invocation

- Symbol names may be up to 16 characters rather than 8, and the symbol name may contain an underscore ("_") character in any character(s) after the first.
- If a symbol name ends with an underscore (not a currently supported symbol character), then its value may be up to 44 characters long.
- The system symbol table may now be up to (approximately) 56K (up from 32K)
- The system symbol service ASASYMBM will be updated to provide a higher-efficiency entry point and path to accomplish substitution so that it will be more suitable to the needs of its exploiters (including JES).

Usage & Invocation - IEASYMxx

- Symbol names up to 16 (was 8) with substitution text length limited to name-length+1. The name may contain "_" in any character position(s) other than the first
- Symbol names up to 16 with a last character of "_" with substitution text length up to 44
- You may not define in the system symbol table two symbols where
  - one of them has no underscores and the name portion is no longer than 8 characters
  - the other begins with the name portion of the first followed immediately by an underscore.

For example, &SYM and &SYM_2 may not both be defined.

Exec ASASYMUN is provided to help locate within JCL and parmlib

Usage & Invocation - IEASYMxx

- You may not define a symbol that begins the same as a reserved system symbol but with an underscore right after the “normal” end of the reserved symbol (for example, since &SYSNAME is a reserved symbol, you may not define &SYSNAME_1).
- When matching a pattern against symbols, an attempt will first be made to match against a symbol with underscores and if that does not succeed, then an attempt will be made to match against a (shorter) symbol without underscores. For example, with just symbol &SYM_2 defined, the string &SYM_2 will be replaced by the value for symbol &SYM_2. But with just &SYM defined, the string &SYM_2 will be replaced by the value for symbol &SYM followed by “_2”.

Usage & Invocation - ASASYMBM

- New entry ASASYMBF
  - User provides 1024-byte work area (saves GETMAIN/FREEMAIN)
- New input option to ASASYMBM / ASASYMBF: SYMBT1
  - Multiple symbol areas in one call
  - Request to preserve alignment (same as JES instream data set processing in z/OS V2R1)
  - No double ampersand (leave them alone, same as JES instream data set processing in z/OS V2R1)
  - Input symbol area was returned by IEFSJSYM service

Usage & Invocation - Commands

- DISPLAY SYMBOLS is updated to display the longer symbol name and longer substitution text value. The columns currently used for the start of the name and the start of the substitution text will change so that a 16-char symbol name with 44-char substitution text will fit on a single line.

Usage & Invocation - ASASYMUN

- Scans a PDS(E) looking for &xxx_ (where &xxx._ would be safe).
- Invoked from ISPF via
  EXEC execdsn(ASASYMUN) 'scandsn'
- Ignores if it thinks the line is from an OBJ (has x'02' in column 1, then TXT, then blank)
- If you have JES symbols with underscores as an instream data set (as of z/OS V2R1) within the data set that you are scanning, those will be shown, but they go through JES symbol resolution so are not a concern. ASASYMUN does not differentiate this case from other JCL situations.
- Outputs:
  - Data set name
  - (for each line identified)
    - Member name, line number
    - The line itself

Migration & Coexistence Considerations

- There will be a migration action for ARM. This is analogous to the migration action that was identified in z/OS 1.4 when the symbol table grew to (approximately) 32K.
- Users of the constants within ASASYMBP representing the maximum size of a symbol name, the maximum size of its symbol value, and the maximum total size of the symbol table will have to change to accommodate the larger sizes of z/OS 2.2.
- APAR OA46739 is compatibility/toleration support for z/OS V1R13 and z/OS V2R1. It is needed if use of long symbol names, or long symbol values, or symbols with underscores is performed on one of those releases on behalf of a z/OS V2R2 system.

Migration & Coexistence Considerations

- Exploiters who relied on the size of a symbol value not exceeding the size of a symbol name will have to change. But it is deemed fully acceptable not to worry about any truncation that might occur and the user “gets what they get”. For example, if there is substitution into an 80-character buffer but the substitution truly needs 100 characters, it is OK to use the characters that did fit into the buffer, just as if the customer had entered those 80 characters without symbols, even if they might not make sense when truncated after column 80.)
- Owners of JCL that might have symbols that are not delimited by “period” that are followed immediately by an underscore do not need to change, but might choose to do so to avoid any confusion. The easiest change is to delimit the symbol by the trailing period. z/OS provides the ASASYMUN exec to help scan all the members of a (for example, JCL or parmlib) PDS(E) to locate any such occurrences.

Presentation Summary

- z/OS V2R2 system symbols may have
  - Name with up to 16 characters
  - Name that contains underscore(s)
  - Value, when name ends with underscore, up to 44 characters (up to you to make sure no line overflow occurs)

Appendix

- Publications:
  - z/OS MVS Programming: Assembler Services Reference ABE-HSP
  - z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN
  - z/OS MVS Initialization and Tuning Reference
  - z/OS MVS System Commands