z/OS V2.5 IBM Education Assistant

Solution Name: 31bit / 64bit interoperability including COBOL / Java

Solution Element(s): z/OS Language Environment

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

Introduction to z/OS LE’s support on 31-bit / 64-bit interoperability including COBOL / Java.

Overview

- Who (Audience)

  As an application programmer, I want to provide interoperability between 31-bit and newly created 64-bit programs to reside on the same TCB.

- What (Solution)

  Language Environment provides support for 31-bit and 64-bit programs interoperability.

- Wow (Benefit / Value, Need Addressed)

  An application architect can integrate 64-bit open eco-systems with existing legacy assets without having to re-architect.

  An application architect can effortlessly do fit-for-purpose placement of the application without having to re-architect the application code.

Usage & Invocation

CEL4RO31 - Prepare and run program in AMODE 31 environment

This CWI is called to prepare and run program in AMODE 31 Language Environment environment from AMODE 64 Language Environment environment.

Syntax

void CEL4RO31(RO31_INFO_PTR)

RO31_INFO_PTR(input/output): A doubleword pointer points to RO31_INFO in a piece of below the bar storage which contains a fixed length structure RO31_CB and three optional variable length parts: RO31_module, RO31_function and RO31_arguments.

Call this CWI interface from C as follows:

```
#include <__le_cwi.h>

void* RO31_info_ptr;
CEL4RO31(RO31_info_ptr);
```

Usage & Invocation

CEL4RO31 - Prepare and run program in AMODE 31 environment

Call this CWI interface from assembler as follows:

```
LG      R6,X’4B8’ Get CEELAA address
LG      R6,88(R6) Get CEELCA address
LG      R6,8(R6) Get CEECAA address
LG      R6,X’448’(R6) Get 64bit fast vector address
LG      R5,8(R6) Get CEL4RO31 environment
LG      R6,8(R6) Get CEL4RO31 routine address
BASR    R7,R6 Branch to CEL4RO31
```

Usage & Invocation

CEL4RO64 - Prepare and run applications in AMODE 64 environment

This CWI is called to prepare and run applications in AMODE 64 Language Environment environment from AMODE 31 Language Environment environment.

Syntax

void CEL4RO64(RO64_INFO_PTR)

RO64_INFO_PTR(input/output): A fullword pointer points to RO64_INFO in a piece of below the bar storage which contains length structure RO64_CB and three optional variable length parts: RO64_module, RO64_function and RO64_argumentsa fixed.

Usage & Invocation

CEL4RO64 - Prepare and run program in AMODE 64 environment

Call this CWI interface from assembler as follows:

```
 L      R15,1024(R12) Get 31bit fast vector address
 L      R15,8(R15) Get CEL4RO64 routine address
 BASR   R14,R15 Branch to CEL4RO64
```

Usage & Invocation

CELQGPIB - Provide Interoperability Parameter Buffer below the bar

This CWI is called to provide Interoperability Parameter Buffer below the bar and return the address of the buffer.

Syntax

void celqgipb(buf_len, ipb_addr, retcode)

buf_len(input): A required parameter, a fullword integer containing the buffer length that the caller passed in.

ipb_addr(outpout): A required parameter, a pointer points to the address of a piece of below the bar storage, or 0 if no storage obtained.

retcode(output): A fullword integer indicating the return code of this service.

Usage & Invocation

CELQGPIB - Provide Interoperability Parameter Buffer below the bar

Call this CWI interface from assembler as follows:

```
LG    R6,X’4B8’ Get CEELAA address
LG    R6,88(R6) Get CEELCA address
LG    R6,8(R6) Get CEECAA address
LG    R6,X’448’(R6) Get 64bit fast vector address
LG    R5,48(R6) Get celqgipb environment
LG    R6,56(R6) Get celqgipb routine address
BASR  R7,R6 Branch to celqgipb
```

Usage & Invocation

- Considerations and restrictions
  - Program Management
    - All AMODE 31 programs involved in the interoperability scenario need to be non-XPLINK.
    - All AMODE 31 and AMODE 64 programs involved in the interoperability scenario need to be reentrant.
    - There is no support for AMODE 24 programs and AMODE 64 programs interoperability.
  - Parameter/return types
    - Complex parameters and return types(ie. a structure or a pointer) are passed without deep copy.
    - AMODE 64 pointers are passed to AMODE 31 programs as is, which can not be dereferenced in AMODE 31 programs.
  - Setting runtime options
    - The secondary Language Environment environment will inherit the runtime option TRAP value and the first sub-option value of runtime option TERMTHDACT of the primary Language Environment environment.

Usage & Invocation

- Considerations and restrictions
  - POSIX behaviors
    - After calling CEL4RO64, the AMODE 31 primary Language Environment environment would not be allowed to call POSIX functions
    - The AMODE 31 secondary Language Environment environment would be forced to be POSIX(OFF) while the AMODE 64 secondary Language Environment environment would be forced to be POSIX(ON)
    - For a multi-threaded AMODE 64 program, only one thread can make AMODE switching calls.
    - An asynchronous signal will be handled only when the AMODE switching thread is executing AMODE 64 programs when it is received.
  - Recovery
    - Language Environemnt will traverse stack frames to find condition handler until AMODE 31 and AMODE 64 environments boundary.
    - Registering ESTAE/ESPIE exit routine after Language Environment in AMODE 31 and AMODE 64 interoperability programs may result in unexpected behaviors

Usage & Invocation

- Considerations and restrictions
  - Termination
    - The normal termination and abnormal termination process would terminate the both Language Environment environments.
    - If an unhandled condition with severity higher than 2 occurred in the secondary Language Environment environment causes the termination, the primary Language Environment environment would be terminated with abend 4094-1024.
  - Performance
    - Transition between AMODE 31 and AMODE 64 programs has more overhead than a normal function call, avoid using it in a performance sensitive spot.
  - Standard stream
    - The redirection of standard streams in the secondary Language Environment environment would result in unpredictable behavior.
    - The redirection of standard streams in the primary Language Environment environment would not impact the standard streams in the secondary Language Environment environment.

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - JNI / JVM

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Sysplex must be at the new z/OS level: No

Installation & Configuration

- None

Summary

z/OS Language Environment provides support for 31-bit and 64-bit programs interoperability.

Appendix

- Language Environment Vendor Interfaces