

# **z/OS V2.5 IBM Education Assistant**

Solution Name: Provide MetalC C header file of IEWBUFF

Solution Element(s): Binder

July 2021



# 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 the new function being delivered for z/OS V2.5 Program Management

# Overview

---

- Who (Audience)
  - Metal C application programmers that calls the binder API
- What (Solution)
  - A new Metal C header of Binder API buffer has been provided in both Unix path `/usr/include/zos/iewbubuf.h` and MVS dataset `SYS1.SIEAHDR.H(IEWBUBUF)`.
- Wow (Benefit / Value, Need Addressed)
  - **Issue:** Currently two kinds of binder API buffer mappings are available to venders (1). Assembler mapping provided in `SYS1.MACLIB(IEWBUFF)`. (2). C/C++ mapping provided in `/usr/include/__iew_api.h`. Some venders are trying to call the binder assemble API from Metal C programs. If a GETx API is called, the binder fills the binder buffer with data retrieved. However, a Metal C user can not use the mapping in `__iew_api.h` directly as this header requires LE environment. In addition, some fields, for example, bits in `BNL_BIND_FLAGS` below, are only defined in the assembler mapping but missed in the C/C++ mapping.
  - **Benefit:** Enable Metal C users, especial those who migrate their program for assembler, to use a C mapping that is exactly identical to the assembler mapping.

# Usage & Invocation(1 of 2)

---

A Metal C sample named getesd.c that utilize this header to access binder API buffer

```
#include <iwbubuf.h>
#include <string.h>

#define ESD_COUNT 1000
#define BUFFER_LEN sizeof(iewApiBuf)+sizeof(iewESDEntryV6)*ESD_COUNT
char BUFFER[BUFFER_LEN];
void main()
{
    iewApiBuf *pHeader;
    pHeader = (iewApiBuf*)BUFFER;
    memcpy(&(pHeader->iewApiBuf_id),"IEWBESD ", 8);
    pHeader->iewApiBuf_leng = BUFFER_LEN;
    pHeader->iewApiBuf_version = 6;
    pHeader->iewApiBuf_entry_leng = sizeof(iewESDEntryV6);
    pHeader->iewApiBuf_entry_count = ESD_COUNT;
    . . .
}
```

# Usage & Invocation(2 of 2)

---

To compile this sample, users may run command xlc in Unix

```
xlc -S -qmetal -qNOSEARCH -I /usr/include/zos -I /usr/include/metal getesd.c
```

If this step succeeds, an assemble source module named getesd.s will be generated.

Next run command xlc to assemble getesd.s.

```
xlc -o getesd.o -c getesd.s
```

If this step succeeds, an object module named getesd.o will be generated.

Finally link this object module to generate a program module.

For example,

```
ld -o getesd getesd.o
```

If this step succeeds, an executable program module named getesd will be generated.

# Interactions & Dependencies

---

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



# Upgrade & Coexistence Considerations

---

- N/A

# Installation & Configuration

---

- N/A

# Summary

---

- Enable Metal C users to use a C mapping that is exactly identical to the assembler mapping.

# Appendix

---

- N/A