IBM Education Assistance for z/OS V2R1

Item: API DLL Datasets Resident Support
Element/Component: Binder

Material is current as of June 2013

© 2013 IBM Corporation

Agenda

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

© 2013 IBM Corporation

Trademarks

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

© 2013 IBM Corporation

Presentation Objectives

- What are binder API DLLs?
- Why data set resident support was needed
- How to use it
- Compatibility with UNIX resident support
- Rollback

© 2013 IBM Corporation

Overview

- Binder provides C/C++ APIs in the form of DLLs. They provide high-level language (LE) applications a more natural means of using the binder regular and fast data access APIs:
  - Introduced in z/OS V1R9
  - XPLINK DLLs added in z/OS V1R12
- Problem Statement / Need Addressed
  - DLLs install into UNIX file system only. Run-time requirement is on end-users of application. Requires LIBPATH (unless explicit load).
- Solution
  - Ship equivalent DLLs also into MVS data sets
  - Needed to also provide matching side-files as well
- Benefit / Value
  - End-users avoid UNIX administration where not required

© 2013 IBM Corporation

Usage & Invocation

- Existing shipped files:
  - Header:
    - /usr/include/__iew_api.h
  - DLLs:
    - /usr/lib/iewbndd.so
    - /usr/lib/iewbnddx.so [XPLINK]
  - Side-files:
    - /usr/lib/iewbndd.x
    - /usr/lib/iewbnddx.x [XPLINK]

© 2013 IBM Corporation

Usage & Invocation

- z/OS V2R1 new shipped parts
  - Header:
    - *No change!* Application provider still needs UNIX.
  - DLLs – SYS1.SIEAMIGE is in default Link List search:
    - SYS1.SIEAMIGE(IEWBNDD)
    - SYS1.SIEAMIGE(IEWBNDDX) [XPLINK]
  - Side-files – Needed because DLL name match is exact:
    - SYS1.SIEASID(IEWBNDD)
    - SYS1.SIEASID(IEWBNDDX) [XPLINK]

© 2013 IBM Corporation

Usage & Invocation

- Batch invocation example

```
// JCLLIB ORDER=(CBC.SCCNPRC)
// SET    SRCLIB=SYS1.SAMPLIB
// SET    MODLIB=&SYSUID..PDSELIB
//*
//CBAPCCC  EXEC PROC=EDCXCBG,
// CPARM='OPTFILE(DD:COPTS)',
// BPARM='OPTIONS(BOPTS)',
// INFILE=&SRCLIB(IEWAPCCC)
//COMPILE.COPTS    DD *
 NOSEARCH,SEARCH(/usr/include)
 RENT,LIST
/*
//*
//BIND.SYSLMOD  DD DSN=&MODLIB(IEWAPCCC),DISP=SHR
//SIDEDECK   DD DSN=SYS1.SIEASID,DISP=SHR
//BIND.SYSIN DD *
  INCLUDE SIDEDECK(IEWBNDDX)
/*
//BIND.BOPTS    DD *
 DYNAM=DLL
/*
//* Go Step will use SYS1.SIEAMIGE(IEWBNDDX)
```


© 2013 IBM Corporation

Usage & Invocation

- UNIX invocation example:

```
export _C89_CSUFFIX_HOST=samplib
c89 -Wc,"dll,lang(extended)" -co iewapccc.o "//'sys1.samplib(iewapccc)'"

export _C89_EXTRA_ARGS=1
c89 -Wl,dll iewapccc.o "//'sys1.sieasid(iewbndd)'"

export LIBPATH=./
./a.out ./a.out # this works proving that LIBPATH setup is not required
```


© 2013 IBM Corporation

Migration & Coexistence Considerations

- Additional compatibility consideration – UNIX links
  - UNIX file system DLLs in /usr/lib now have (hard) links to all uppercase names matching the MVS data set members:
    - iewbndd.so ↔ IEWBNDD
    - iewbnddx.so ↔ IEWBNDDX
  - When binding with the side-files from SYS1.SIEASID, if the application is run POSIX(ON), binder API DLL will be found in the UNIX file system (as in the prior example).

© 2013 IBM Corporation

Migration & Coexistence Considerations

- Rolled back into z/OS V1R13:
  - APAR OA39387 PTF UA65826
  - Does not add the UNIX links

© 2013 IBM Corporation

Installation

- Must (re-)bind to use new DLLs

- UNIX installed DLLs and MVS data installed DLLs are separate files
  - UNIX DLLs are not external links

© 2013 IBM Corporation

Appendix

- z/OS MVS Program Management: Advanced Facilities – SA22-7644
  - 5.1 Using the binder C/C++ API and headers
  - SA22-7644-14 is the z/OS V1R13 *refresh* which includes information for APAR OA39387

© 2013 IBM Corporation