

# *Automating Remote DB2 Binds Using CA Endeavor® Software Change Manager*

**CA World 2008**

**Focus Area: Mainframe and Multi-Platform Application  
Development**

**Session Code: MC107SN**



**Rose A. Sakach**

**Endevor Practice Leader - RSH Consulting, Inc.**

**R.Sakach@RSHConsulting.com - 617.969.9050 - [www.rshconsulting.com](http://www.rshconsulting.com)**

# Abstract

In a typical DB2 programming environment, a DB2 bind invocation is required whenever the DB2 source code has been modified. This session will demonstrate the benefits of utilizing CA Endeavor® SCM to automate the DB2 bind function. In addition, we will examine how to achieve automation when the DB2 subsystems associated with specific life cycle stages reside on multiple mainframe system images.

Topics include determining your DB2 environment, creating and utilizing CA Endeavor SCM symbolic parameters, and options for coding DB2 bind statements. Sample CA Endeavor SCM processors and symbolics will be provided.

# *Agenda*

- **Why automate**
- **Terms and process review**
- **Information gathering**
- **Mapping life cycles**
- **Connecting DB2 subsystems**
- **Processor considerations**
- **Communication**

# *Why Automate*

- **Streamline the process**
- **Reduce risk**
- **Improve quality**
- **Cost savings**
- **Why not**



# *Terms and Process Review*

- **DB2 Subsystem id (SSID)**
  
- **DB2 Remote location id**
  
- **DB2 pre-compile or DB2 coprocessor**
  - **SQL Syntax validation**
  - **Host language replacement**
  - **Consistency token**
  - **Version parameter**
  - **Data base request module (DBRM)**

# *Terms and Process Review*

## ■ DB2 bind

- Package
- Collection
- Plan

## ■ DB2 Stored Procedures

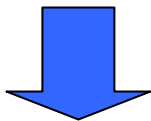
## ■ WLM Address Space

# Terms and Process Review

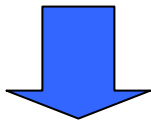
DB2 Precompile



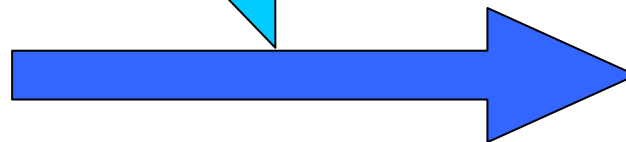
Source Compile



Object Link Edit

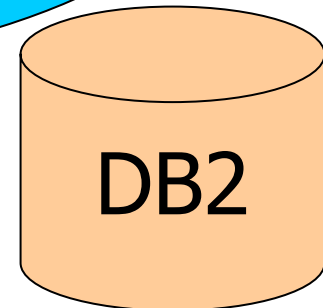


DB2 Bind



**Note:**

*The DB2 coprocessor COBOL option eliminates the need for the pre-compile step*



# *Information Gathering*

## *DB2 considerations*

- **Plan Bind Requirements and Coordination**
  
- **Package Bind type definitions**
  - Packages = programs (SQL)
  - Collections = group of packages
  - Plan = group of collections
  
- **Bind card values (bind parameters)**
  - Enterprise standards vs. application standards
  - DB2 Application / CA Endevor SCM inventory alignment
    - Package and collection naming standards
    - Qualifiers - test table use
    - Owner

# Information Gathering

## *DB2 considerations*

- **Bind card overrides**
  - DBA only vs. end-user
- **Back-binding**
  - Tables mapped to prior CA Endeavor SCM stage
- **Versioning**
  - None vs. AUTO vs. something else
- **Security and administration**
  - DB2 or RACF
  - Alternate id authority
    - Understand and mitigate the risk
  - Access grants

# *Information Gathering*

## *DB2 considerations*

- **Load module back-out process**
  - Do the binds need occur?
- **Emergency fix process**
  - How do binds occur?
- **To connect or not to connect**
  - Production LPAR off limits?
- **To automate or batch**
  - **Coordination with other integrated processes**
    - **Change Request System**
    - **DB2 table modifications**
    - **Production Build Schedule**

# *Information Gathering*

## *DB2 considerations*

### ■ **Stored Procedure considerations**

- **CA Endeavor SCM identification / process**
  - **SP Naming Standards**
  - **WLM Configuration**
- **Implementation process**

# Information Gathering

## ■ Fundamental DB2 Data\*

- Total number of DB2 subsystems
- Where they reside (LPAR)
- When/Where binds are performed
- Required authority
- Who has the authority to bind
- Use of versioning
- Bind parameter values
- DB2 component naming standards
- Exceptions



\* Be sure to obtain the enterprise view prior to finalizing design

# Information Gathering

For Application: **AAA**

Endevor Stage ==>	DEV	TEST	QA	EMER	PROD
LPAR					
DB2 Subsystem Id					
DB2 Remote Location Id					
Owner					
Qualifier					
Collection Id					
DB2 Authorization Id					

# *Information Gathering*

## *CA Endeavor SCM considerations*

### ■ End-user

- DB2 expertise
- Ease of use
- Flexibility

### ■ Administrator maintenance

- Minimize processor groups
- Maximize end-user control

### ■ Processor philosophy

- Type vs. processor group distinction

### ■ Processing philosophy

- Precompile - Compile - Link - Bind

# Information Gathering

## *CA Endeavor SCM considerations*

### ■ New types

- DB2 dclgens, plans, and tables
  - Intuitive type names
  - Distinction by DB2 subsystem
- Migration requirements
  - Life cycle variations (for new DB2 types)
  - Approval considerations (DBA?)

### ■ DBRM library ownership

- Convert to Endeavor naming standard

### ■ When to bind

- DB2 program action
  - Add / Update / Generate
  - Move
- Other?

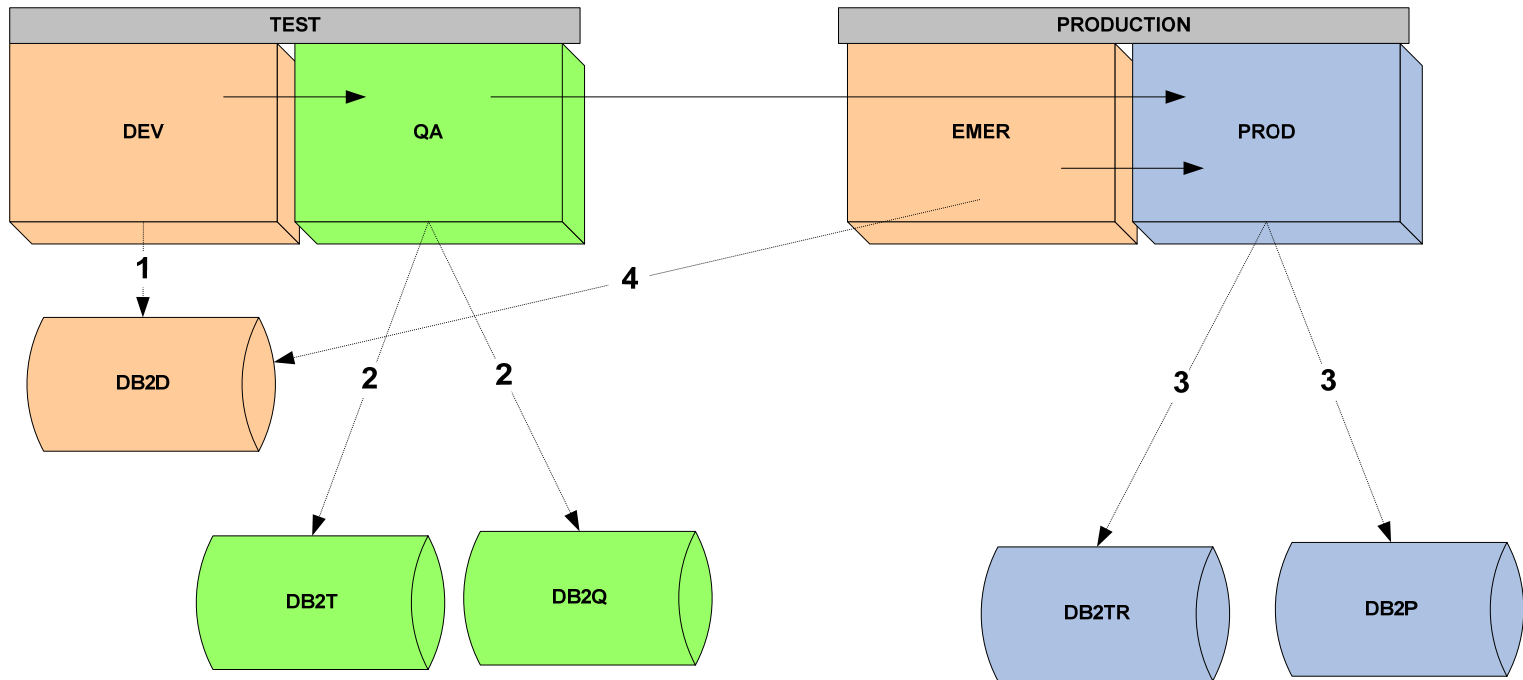
# *Information Gathering*

## *CA Endeavor SCM considerations*

- **Alternate user id**
  
- **Program back-out**
  - **Aligning current process with DB2 requirements**
  
- **Emergency Fix**
  - **Aligning current process with DB2 requirements**

# Mapping Life Cycles

Sample 1: Endeavor SLC – DB2 Database Configuration (No GOMs)



**1 - DB2 programs ADDED to DEV invoke package bind to DB2D**

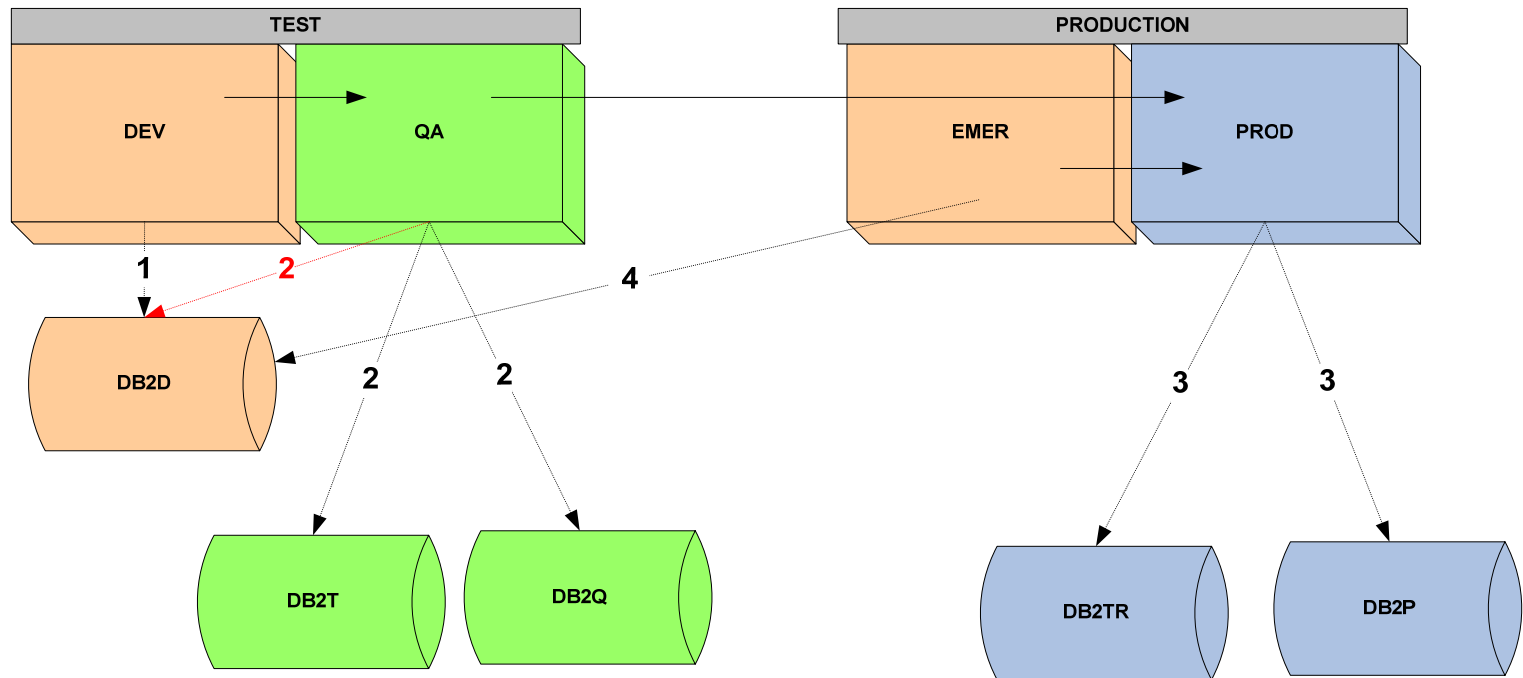
**2 - DB2 programs MOVED to QA invoke package bind to DB2T and DB2Q**

**3 - DB2 programs MOVED to PROD invoke package bind to DB2P and DB2TR**

**4 - DB2 programs ADDED to EMER invoke package bind to DB2D**

# Mapping Life Cycles

Sample 2: Endeavor SLC – DB2 Database Configuration (DEV-to-QA GOM)



1 - DB2 programs ADDED to DEV invoke package bind to DB2D

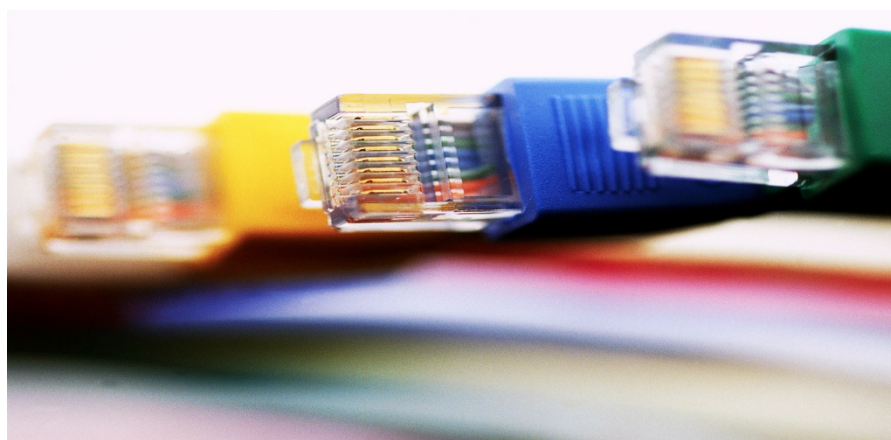
2 - DB2 programs MOVED to QA invoke package bind to DB2T and DB2Q AND a bind to **DB2D**

3 - DB2 programs MOVED to PROD invoke package bind to DB2P and DB2TR

4 - DB2 programs ADDED to EMER invoke package bind to DB2D

# Connecting DB2 Subsystems

- CA Endeavor SCM must reside on an LPAR with a DB2 subsystem
- Distributed data facility (DDF) must be configured for DB2
- Bind syntax (for local or remote)
  - DB2 remote location id replaces SSID in the bind syntax
    - `BIND PACKAGE (REMOTE LOCATION ID.COLLECTION ID)`
  - Continue to use the SSID in the DSN statement



# Processor Considerations

## ■ Building bind cards

- Processing combinations (i.e., static bind cards, CA Endeavor SCM symbolics, external tables, program interfaces)

## ■ CA Endeavor SCM “smart” symbolics a.k.a. ZCONCAT

## ■ CA Endeavor SCM site symbolics

- DB2 symbolics
- DB2 library names (i.e., DSNLOAD)
- Bind parameters

## ■ CA Endeavor SCM includes

- Bind code utilized within more than one processor

# Processor Considerations

## ■ Smart Symbolics

```
/**          DB2 subsystem / LPAR XREF
/**          -----
/**          390A = DB2P/DB2TR
/**          390B = DB2T/DB2Q
/**          390C = DB2D
/**          390D = DB2Z
/**-----
/**Primary connecting DB2 Subsystem = DSNZ on 390D
/**Alternate connecting DB2 subsystem = DSND on 390C
/**-----
//          DB2SYS=' &D2S&MVSID' ,
/**
//          MVSID=' 390D' ,
/******
//          MVSID=' 390C' ,
//          D2S390C=' DB2D' ,
//          D2s390D=' DB2Z `
//
```

# Processor Considerations

## ■ Smart Symbolics

```
/**
/**          DB2 Qualifiers
/**-----
//          DB2QUAL = '&DB2Q&C1SYS(1,3,0)',
/**-----
/**          Application = AAA
/**          -----
//          DB2QAAA = '&D2QAAA&C1STGID',
//          D2QAAA1 = 'DEVL',
//          D2QAAA2 = 'QUAL',
//          D2QAAAP = 'PRDN',
//          D2QAAAT = 'TRNG',
```

# Processor Considerations

## ■ Site Symbolics

<b>SYMBOL</b>	<b>VALUE</b>
<b>#DSNLOADD</b>	<b>SYS2 .DB2D .DSNLOAD</b>
<b>#DSNLOADT</b>	<b>SYS2 .DB2T .DSNLOAD</b>
<b>#DSNLOADQ</b>	<b>SYS2 .DB2Q .DSNLOAD</b>
<b>#DSNLOADP</b>	<b>SYS2 .DB2P .DSNLOAD</b>
<b>#DSNLOADZ</b>	<b>SYS2 .DB2Z .DSNLOAD</b>

# Processor Considerations

## ■ Include

```
++include DB2BIND
/*****
/* Processor INCLUDE
/* Name: DB2BIND
/* DESC: CONWRITE or GEN BIND cards; execute BIND
/*****
//GENBINDC EXEC PGM=IEBGENER,
//      MAXRC=0,
//      COND=(0,EQ,RETBIND)
//SYSUT2 DD DSN=&&BND CARDS,DISP=(OLD,KEEP,DELETE)
//SYSUT1 DD *
  DSN SYSTEM(&DB2SYS)
  BIND PACKAGE (&DB2RML..&DB2CID) -
      MEMBER (&C1ELEMENT) -
      OWNER (&DB2OWN)          QUALIFIER (&DB2QUAL) -
      SQLERROR (&DB2SQLE)      VALIDATE (&DB2VAL) -
      ISOLATION (&DB2ISO)      RELEASE (&DB2REL) -
      ACTION (&DB2ACT)         EXPLAIN (&DB2EXPL)

END
```

# Processor Considerations

## ■ The Bind Step

```
//BIND EXEC PGM=IKJEFT01,MACRC=4,  
// COND ((4,LT,RETBIND) ,(4,LT,GENBIND) ,(0,LT,SWAPOUT) )  
//*  
//STEPLIB DD DSN=&#DSNLOADZ,DISP=SHR  
//*  
//DBRMLIB DD DSN=&DBRMLIB,DISP=SHR  
//SYSTSPRT DD DSN=&&BINDLIST,DISP=SHR  
//SYSTSIN DD DSN=&&BNDCARDS,DISP=(OLD,DELETE)
```

---

**Note: IKJEFT01: Returns CC=0 if one step CC=0 regardless of others**  
**IKJEFT1B: Returns highest RC**

# Processor Considerations

Invoking the bind:

- Entry stage **GENERATE**
- Entry stage **MOVE**
- Subsequent stage **MOVE** or **GENERATE (GOM)**



# Processor Considerations

## Entry stage **GENERATE**

- Allocate additional &TEMP files (I.e. Bind listing file, bind card parm file)
- Ensure DB2 pre-compile (or compile if using DB2 coprocessor) step outputs to CA Endeavor SCM DBRM library
- Retain existing precompile, compile, and link edit steps; append BIND steps (assuming precompile-compile-link-bind)
- Use appropriate DB2 DSNLOAD libraries

# Processor Considerations

## Entry stage **GENERATE**

- **BIND01 – retrieve user specified card**
  - Use CONWRITE utility (C1BM3000 retrieve will work at a cost!)
  - Specify “no signout” and “search” (if applicable)
  - Ensure type is defined
  - Write element to temp file (&&BNDCARDS)
  
- **BIND02 – generate application default bind card**
  - Bypass if BIND01 CC=0
  - IEBGENER into temp file (&&BNDCARDS) OR programmatically read predefined application/DB2 parameter file and write out to temp file
  - Use in-stream data on IEBGENER to utilize Endeavor symbolic values

# Processor Considerations

## Entry stage *GENERATE*

- BIND03 – swap out user id for alternate id
  - For use in DB2 address space (ASXBUSER)
- BIND04 – execute the bind (IKJEFT01 or IKJEFT1B)
  - SYSTSIN=&&BNDCARDS
  - SYSTSPRT=bind listing library
- BIND05 – swap alternate id back to user id
- Add bind listing file to existing store/print listing step

# Processor Considerations

## *Entry / subsequent stage* **MOVE**

- Allocate additional &TEMP files (i.e. bind listing file, bind card parm file)
- Retain existing library copy steps; append BIND steps
- Insert BIND01 – BIND05 steps as coded in *entry stage* **GENERATE** (May require additional bind if GOM)
- BIND06 – copy DBRM file to next stage
- Add bind listing file to existing store/print listing step
- BIND07 – copy bind listing file to next stage
- REMINDER - retain MOVECL (BC1PMVCL) as final step (if monitoring components)

# Processor Considerations

## Special **GENERATE?** (Don't do it!)

- Create a special generate processor to execute bind at non-compile stage in the SLC.
  - **CAUTION!** MCF entries are modified (LAST GEN DATE) and can cause component validation errors, footprint exceptions, and will create a new component list level.  
**NOT RECOMMENDED.**



# *Processor Considerations*

## **Implementation Tips**

- **Ensure the design meets the requirements**
  
- **Ensure all applications will work within the design**
  
- **Test all features at all stages**
  - **Engage users and DBAs**
  - **Use your “dummy” id**

# Processor Considerations

## Implementation Tips

### ■ Processor and processor group considerations

- Utilize new processor groups for testing
- Create new or use existing processors?
  - Consider risk to production
  - Consider implementation approach - phased or enterprise wide?
  - Consider resource and time constraints during implementation and back-out
  - **\*CAUTION\*** new processor would result in C1G0147E for an element moving up the SLC: THE GEN PROCESSOR FOR TYPE *element-type* (old-processor) WAS NOT THE LAST PROCESSOR EXECUTED (new-processor)

### ■ Prepare to back-out

- Have a plan
- Code batch admin JCL in advance

# Communication

- **Establish CA Endeavor SCM / DB2 partnership**
  - **Make the project a collaboration**
  - **Engage DB2 resources early and often**
  - **Strengthen during data gathering**
  - **Develop procedures for handling DB2 release upgrades**



- **Prepare your users**
  - **Make project details and timelines accessible**
  - **Prepare and provide end-user and DBA training**
  - **Provide documentation**
    - **Sample output**
    - **Describe common errors, document message codes, provide resolutions**
    - **Contact information**

# Communication

## ■ Sample FAQ entry:

Qxx. I submitted a job to ADD/UPDATE/GENERATE my program in Endeavor and the DB2 BIND step fails with RC=12. Do I need to contact Endeavor Administration to correct this error? The messages issued by DB2 look like these:

READY

DSN SYSTEM(DSNZ)

DSNZ NOT VALID SUBSYSTEM ID, COMMAND TERMINATED

*bind card parameter information is here.....*

READY

COMMAND BIND NOT FOUND

READY

END

READY

**Axx. Not necessarily.** First CHECK TO BE CERTAIN your Endeavor job ran on 390D. The auto binds are currently setup to connect to the DB2 subsystem on 390D. If you try to run your job on 390A, B or 390C, the bind step will fail with the above error message. If you ran your job on 390D – contact *your-service-desk @ xxx-xxx-xxxx* for assistance.

# *In Summary*

- **Yes you can automate DB2 binds and the DB2 remote package bind feature can be utilized if CA Endeavor SCM does not reside on the same system image as the “binding” DB2 subsystem**
- **Learn where the applications bind throughout the SLC; and ensure the new process fits the production migration schedule**
- **Do your homework; weigh the many options; analyze the best way to implement all applications; fit the design and meet the requirements**
- **Make training, communication and the CA Endeavor SCM/DB2 partnership an ongoing priority**



# Related Sessions

- **MC108SN – 11/19 – Marco Polo 707 – 8:30am**
  - **Installation and Maintenance with CA Endeavor SCM r12**
- **MC109SN – 11/19 – Marco Polo 707 – 9:45am**
  - **CA Endeavor SCM and the CSV File**
- **MC110SN – 11/19 – Marco Polo 707 – 11:00am**
  - **Know your CA Endeavor SCM Relationships**

# ***“Mix and Munch”***

## ***Networking with Your Peers***

**Join your CA Endeavor SCM peers and CA to mingle and network over lunch.**

In this informal setting, engage in CA Endeavor SCM solution discussions, share ideas, and establish new contacts. A boxed lunch will be provided. Seating is limited and will be on a first come, first served basis.

**Where: San Polo 3405-6, 3505-6**

**When: Wednesday**

**Time: 12:00pm - 1:15pm**



# Related Sessions

- **MC112PN – 11/19 – Marco Polo 707 – 1:15pm**
  - **A Day in the Life an Endeavor Administrator**
- **MC114SN – 11/19 – Marco Polo 707 – 2:45pm**
  - **Global usage of CA Endeavor SCM at Volvo**
- **MC111SN – 11/19 – Marco Polo 707 – 4:00pm**
  - **Improve Quality Of Change Implementations By Integrating Problem And Change Management Processes**
- **MC115SN – 11/19 – Marco Polo 707 – 5:15pm**
  - **CA Endeavor PLC/User Group meeting**
- **MC113SN – 11/20 – Marco Polo 702 – 9:45am**
  - **Concurrent Action Processing – How Fast Can You Go?**

# **Exhibition Center**

## **Related CA and Partner Technology**

### **■ CA**

- **Booth 425 — Mainframe**
- **Booth 549 — Manage**

### **■ Exhibition Center Tours**

- **Sign up at the Info Desk in the Exhibition Center**

