Beyond the Software Life Cycle
CA-Endevor Facilitates Ad-hoc Job Processing

Southern CA Endevor User Group

June 2008

Rose A. Sakach

Endevor Practice Leader - RSH Consulting, Inc.

R.Sakach@RSHConsulting.com - 617-969-9050 - www.rshconsulting.com

Abstract

Beyond the Software Life Cycle
CA-Endevor Facilitates Ad-hoc Job Processing

The current demand for maintaining a high level of production availability can often times require ad-hoc production file updates - a process that is typically associated with time and resource constraints, risk and potential security exposures.

This session will describe a method of streamlining the production file update process by taking advantage of CA-Endevor's ability to store and track JCL changes, control dataset access via batch jobs, enforce code reviews and approvals, provide an audit trail and interface with the organization's scheduling software (i.e. CA-7; ASG's ZEKE). Procedural guidelines and sample Endevor processors will be provided.

AGENDA

Production File Updates

Current Process Issues

Endevor Can Help

Procedural Guidelines

Process Suggestions

RACF Security Recommendations

Monitoring

Production File Updates

Read access to a Production file

- **Ad-Hoc reporting**
- **Dumping data to research and resolve a problem**
- **Copying data to produce valid test cases for testing / training**
- **Emergency backup**

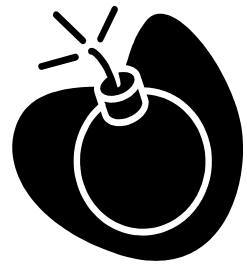
Update access to a Production File

- **Initial file load (new application installation)**
- **File conversion**
- **Emergency system outage (i.e. *firecall*) requires corrupt data repair – remove/replace invalid records**
- **Emergency batch job restart / rerun (outside of scheduling software)**

Current Process Issues

Good production file security policies limit access

Users responsible for resolving the issue typically do not have standing access to the applicable files



Most emergency file update processes:

- **Are typically batch jobs executed outside of the scheduling software**
- **Do not enforce JCL validation and site specific process standards**
- **Are not standardized even though they occur repeatedly**
- **Often bypass formal authorization procedures**
- **Typically permit cart-blanche access resulting in increased risk**
- **Are not automatically tracked and thus not easily auditable**
- **Can result in intolerable outages and audit exposures which may cost the company lots of $$$$$**

Endevor Can Help

JCL repository

Change tracking tool

Mandates review and approvals for JCL changes / migrations

Capable of interfacing with various software products

Standardizes repeatable processes

Can automatically enforce JCL and site security standards

Provides Auditing capabilities

Can you see where this is going?

Endevor Can Help

<u>Production File Update Process</u> (Today)

User obtains authorization, then logs in to Production site and directly updates a sensitive file OR user creates a batch job to perform the file modifications

- **May use generic "firecall" ids with lots of authority**
- **Unrestricted access**
- **No audit trail of who, what, when, how data was modified**
- **No enforceable review of changes**
- **Possible but unlikely audit trail of access request submitted to security**
- **High risk to sensitive data**
- **Open door to regulatory (i.e. SOX) and compliance issues**

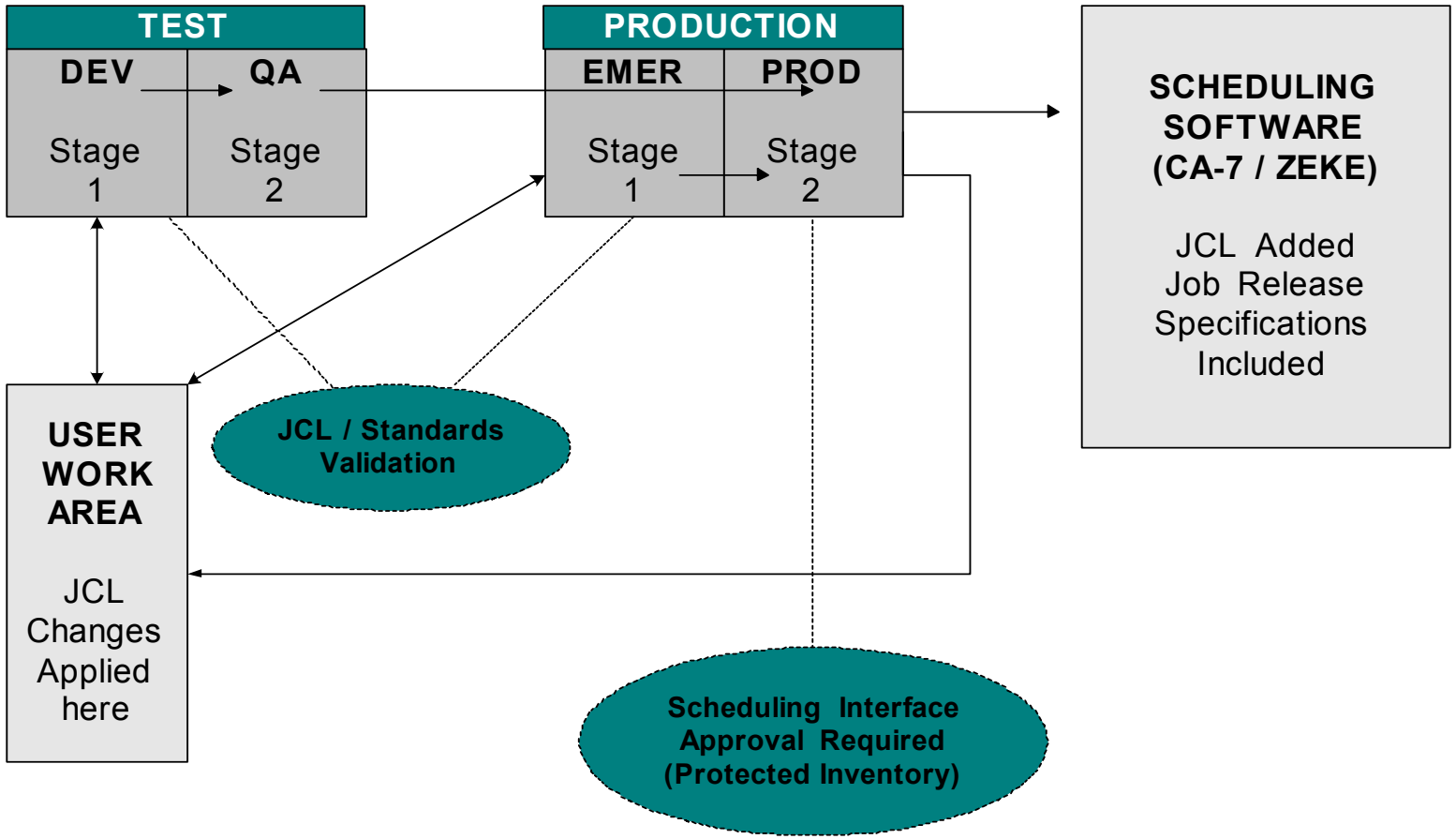
<u>Production File Update Process</u> (Using Endevor)

Authorized ENDEVOR user creates batch job, adds the item to Endevor, migrates to Production via a controlled process

- **Authorization is predefined and controlled**
- **Complete, automatic audit trail**
- **Job must be reviewed/approved prior to migrating to production**
- **Sensitive data file authorization is required and enforced via automation**

Endevor Can Help

Endevor Ad Hoc Job (Special Job) Process



Procedural Guidelines

Production File Update Process – Using Endevor

Step 1

- User RETRIEVES one of the predefined SPECJob type members
  - Note: Appropriate access to data files specified within the job must be granted to the USERID assigned to the job
- User modifies the JCL to perform the required action
- User ADDS the SPECJob member to Endevor entry stage
- User reviews the LISTING library entry for any syntax or site standards coding errors that would cause the element to be marked *FAILED*

Step 2

- User MOVES the SPECJob element to the next stage in the life cycle in preparation of final review and job scheduling considerations

Procedural Guidelines

Production File Update Process – Using Endevor

Step 3

- User CREATES a package to MOVE the SPECJob element to PRODUCTION
  - Note: User can determine whether or not to utilize the Endevor execution window – may be helpful if the job is to be released immediately by the scheduling software (be CAUTIOUS about this!)
- User CASTS package containing the SPECJob element
  - Note: Depending upon the inventory location, element type and package type, this package will require approval from (should be a minimum of two): the data owner (or team lead), a scheduling administrator, a team member or alternate team lead. "Firecall" situations should utilize Emergency type package processing and must follow procedures and obtain approvals according to existing policies.
- Endevor Package SWEEP job process executes the package

Procedural Guidelines

Production File Update Process – Using Endevor

Step 4

- During package processing, the Endevor MOVE processor writes (loads) appropriate records to the scheduling software database
  - Note: Process specifications determine whether or not to place jobs on "HOLD" etc.
- Scheduler (i.e. CA-7 or ZEKE) processes the job scheduling request
- User verifies job execution post Endevor package execution (i.e. Package Status = Executed)

Procedural Guidelines

Specifications to Consider:

- Naming standards for reserved SPEC jobnames: AAA$SP01 – 99
- Ability to VALIDATE JCL requirements (existing software tools?)
  - Jobnames
  - Account Codes
  - USERIDs assigned to the job
  - Dataset names (i.e. no USERID datasets allowed; Application HLQ restrictions)
  - Abend or user-to-be-contacted instructions
- Ability to VALIDATE security requirements
  - USERID naming standard is critical (AAA#SP01 = Read access to AAA)
  - Predefined security permissions
  - Predefined security procedures requesting additional access

Procedural Guidelines

Specifications to Consider:

- **Flexibility in scheduling and data access**
  - **SPJOB type = Release Immediately**
  - **SPJOBH type = Place job on "HOLD"**
  - **SPJDB2 type = Access DB2 data; Release immediately**
  - **Element Comments = Job release instructions (Scheduler reviews these prior to performing approval)**
- **Endevor Sweep Process**
- **Existing Security Structure**
  - **Avoid adding the SPJ ids to existing groups ("willy-nilly")**
  - **Consider auditing the SPJ ids (RACF = UAUDIT)**
- **Culture (Consider user's ability)**

Procedural Guidelines

Implementation guidelines:

- Determine best approach based upon Endevor inventory structure
  - Single SYSTEM w/ 100's of Subsystems?
  - Sweep job in place?
  - Alternate USERID in place? (Guess who needs access to update the Scheduling database?)
- Define TYPES and allocate Endevor libraries
- Consider package id naming standard
- Work out Security Issues in advance
  - Define the SPJ ids specifically for this process
  - Grant appropriate access in advance (may require new profiles to be defined)
    - Application Test file HLQ
    - Application Prod file HLQ
    - Interfacing Application Test HLQ?
    - Interfacing Application Prod HLQ?
  - Plan and Implement process for requesting additional access

Procedural Guidelines

Implementation guidelines:

- **Exploit Validation Possibilities**
  - **JCL checking software (i.e. ASG-JCLPREP or CA-JCLCHECK)**
  - **Rexx Routines**
  - **COBOL or Assembler**
- **Gather scheduling requirements**
  - **Data – record formats, field values etc.**
  - **Security – requirements to interface with CA-7 or ZEKE**
- **Setup automatic reporting process for Audit**
  - **Endevor Reports (by type, by package id)**
  - **Data Security Reports (by SPJ ids)**
- **Provide "how-to" documentation for users**

Process Suggestions

Entry Stage GENERATE

- Insert your JCL validation software here
  - Tip: Code two steps – 1 for JCL validation, 1 for site standards
  - Tip: Code a flag for each which will permit a bypass (as a failsafe)
  - Tip: Store any JCL “rules” within Endevor to take advantage of change tracking
- Utilize CONWRITE to capture and store the JCL in a temporary file
- Store validation listings (CONLIST) within Endevor
- Ensure validation condition codes (issued from the validation software) appropriately mark element add as *FAILED*

Process Suggestions

Entry Stage GENERATE

(Syntax Check Sample Steps)

```
//*************************************************************
//*STEP FUNCTION: EXTRACT THE SPECJob JCL FROM ENDEVOR SM LIBS
//*************************************************************
//CONWRITE EXEC PGM=CONWRITE,PARM='EXPINCL(&EXPINC)',MAXRC=0
//ELMOUT   DD  DSN=&&SPJCL(&C1ELEMENT),
//             UNIT=SYSDA,
//             SPACE=(TRK,(p,s,d),RLSE),
//             DISP=(NEW,PASS,DELETE),
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
```

(Continued on next slide)

Process Suggestions

Entry Stage GENERATE

(Continued from previous slide)

```
//*******************************************************
//*  STEP FUNCTION: Invoke JCL Check for SYNTAX check
//*******************************************************
//JCLPREP EXEC PGM=JCLPREP,COND=(0,LT,CONWRITE),
//            MAXRC=4, EXECIF=(&SYNCHECK,EQ,'Y')
//SYSUDUMP DD SYSOUT=*
//DDIN     DD DSN=&&SPJCL,DISP=(OLD,PASS)
//DDOUT    DD DUMMY
//DDXEFI   DD DSN=&RULESLIB(&RSPJSYN),DISP=SHR
//DDOPT    DD DSN=&JOPTSLIB(&OSPJSYN),DISP=SHR
//DDRUN    DD *
PDS
INCLUDE &C1ELEMENT
/*
//DDWORK1  DD DSN=&&WORK01,
//             DISP=(OLD,PASS)
//DDWORK2  DD DSN=&&WORK02,
//             DISP=(OLD,PASS)
//DDRPT    DD &&SYNLIST,DISP=(OLD,PASS)
//DDXEFP   DD DUMMY
//DDXEFW   DD DUMMY
```

Process Suggestions

Production Stage MOVE

- Prepare JCL for scheduling interface here
  - Tip: Code a program to format JCL and ensure scheduling restrictions are validated
  - Tip: The output of this program will serve as input to the scheduling batch utility
- Utilize CONWRITE to capture and store the JCL in a temporary file
- Code the scheduling batch utility interface here
  - Tip: Ensure the ALTERNATE USERID has appropriate access to load and schedule the job
- Store validation and utility listings (CONLIST) within Endevor
- Ensure target stage is protected (i.e. PACKAGE is required)

Process Suggestions

Production Stage MOVE

(Sample Scheduling Interface)

```
//*******************************************************
//*STEP FUNCTION: Obtain Element Source from BASE/DELTA
//*******************************************************
//CONWRITE EXEC PGM=CONWRITE,PARM=,MAXRC=0
//ELMOUT   DD  DSN=&&SOURCE,
//             DISP=(,PASS,DELETE),UNITE=&UNIT,
//             SPACE=(TRK,(p,s),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//*
//*******************************************************
//*STEP FUNCTION: Execute SPCjob Data format program
//*******************************************************
//SPECCBL  EXEC PGM=SPECCHK,
//         PARM=(&C1ELEMENT,&C1USERID,&C1ELTYPE)
//INFILE   DD DSN=&&SOURCE,DISP=(OLD,DELETE)
//OUTFILE  DD DSN=&&SPECJCL,
//            DISP=(,PASS,DELETE),UNIT=&UNIT,
//            SPACE=(TRK,p,s),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

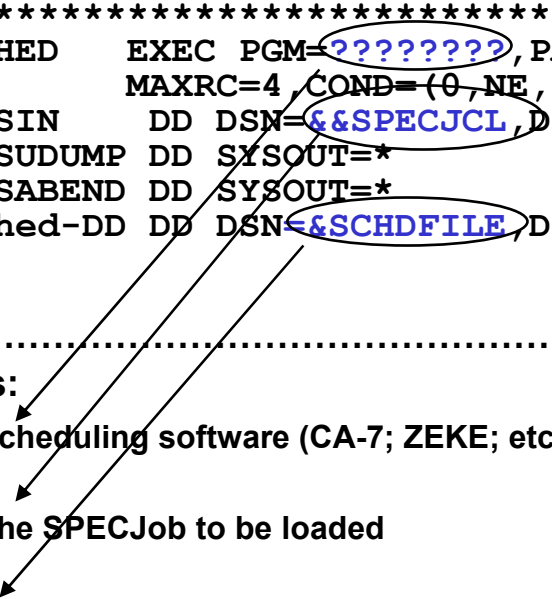
(continued on next slide)

Process Suggestions

Production Stage MOVE

(continued from previous slide)

```
//*************************************************
//*STEP FUNCTION: Execute Scheduling Batch
//*               Interface Program
//*************************************************
//SCHED   EXEC PGM=????????,PARM='parms',
//        MAXRC=4,COND=(0,NE,SPECCBL)
//SYSIN    DD DSN=&&SPECJCL,DISP=(OLD,DELETE)
//SYSUDUMP DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//sched-DD DD DSN=&SCHDFILE,DISP=SHR
```



Notes:

- Scheduling software (CA-7; ZEKE; etc.) batch interface utility program
- The SPECJob to be loaded
- The Scheduling Database / master file

RACF Security Recommendations

Setting up batch USERIDs for the SPEC job card

Sample JCL

```
//AAA$SP00 JOB 99999,'Copy Test Data',CLASS=A,REGION=6144K,
//              MSGLEVEL=(1,1),MSGCLASS=P,USER=AAA#SP01
//STEP001  EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=A
//SYSUDUMP DD  SYSOUT=V
//INDD     DD  DSN=AAA0P.MASTER.D0TRAN.MS00PE1,DISP=SHR
//*
//OUTDD    DD  DSN=BBB00P.ACCNTR.D0BULK.MS00PE1,DISP=SHR
//SYSIN    DD  DSN=PCE00P.AAA0AAA0.CNTL(FIXIT),DISP=SHR
```

Notes:

USERID requirements:

- **UPDATE access to application BBB production files**
- **READ access to application AAA production files**

RACF Security Recommendations

Batch USERID Naming Standard

Sample Format = aaa#SPnn

where:

- aaa = The Application identifier
- #SP = Constant value to indicate SPECJob
- nn = Numeric value 00 – 99

nn Samples:

- 00 = UPDATE or ALTER access to aaa application PROD files; READ access to aaa application TEST files
- 01 = same as 00 + UPDATE access to interface application BBB
- 02 = same as 00 + UPDATE access to interface application CCC
- 99 = same as 00 + UPDATE access to PAY

RACF Security Recommendations

Defining Batch USERIDs

```
AU AAA#SP00 OWNER(AAA) NOPASSWORD NAME('ENDV SPJ')      DFLTGRP(AAA)
AU AAA#SP01 OWNER(AAA) NOPASSWORD NAME('ENDV SPJ-BBB') DFLTGRP(AAA)
AU AAA#SP99 OWNER(AAA) NOPASSWORD NAME('ENDV SPJ-PAY') DFLTGRP(AAA)

ALU AAA#SP00 UAUDIT
ALU AAA#SP01 UAUDIT
ALU AAA#SP99 UAUDIT
```

Notes:

- **Determine typical access requirements – review sampling of prior firecall activities via incidence reporting**
- **Determine if one or many ids are required – potential need, risk etc.**
- **Ensure ALL batch ids are predefined and have the PROTECTED attribute**
- **Specify UAUDIT on all batch ids for reporting**

RACF Security Recommendations

Granting Permission to Data Files

Things to keep in mind:

- Naming standards associated with production vs. test files (HLQ may not indicate)
- Be aware of existing RACF profiles protecting production and test data files (access to several profiles may be required depending upon security standards, file sensitivity etc.)
- When granting access to interfacing application data files, utilize the most specific profile possible (this may prompt a new profile definition but will ensure permission is explicit and access is limited)
- Ensure access level is appropriate (i.e. file creates/deletes require ALTER access)

RACF Security Recommendations

Granting Permission to the Data Files

```
PE  'AAA00P.**'                          ACCESS(UPDATE)  ID(AAA#SP00)
PE  'AAA00T.**'                          ACCESS(READ)    ID(AAA#SP00)

PE  'AAA00P.**'                          ACCESS(UPDATE)  ID(AAA#SP01)
PE  'AAA00T.**'                          ACCESS(READ)    ID(AAA#SP01)
PE  'BBB00P.ACCNTR.D0BULK.MS00PE1'       ACCESS(UPDATE)  ID(AAA#SP01)
              OR
    'BBB00P.ACCNTR.D*.**'

PE  'AAA00P.**'                          ACCESS(UPDATE)  ID(AAA#SP99)
PE  'AAA00T.**'                          ACCESS(READ)    ID(AAA#SP99)
PE  'PAY0P.ACCNTR.B0ATBL.*'              ACCESS(UPDATE)  ID(AAA#SP99)
```

Notes:

- **"00" user is granted access to Production and Test files for the specified application**
- **"01" user is granted the same access as the 00 user in addition to access to the BBB application (historical records have shown instances when AAA application requires update access to BBB production files)**
- **"99" user is granted update access to the PAY application production files in addition to having the same access as user "00"**

RACF Security Recommendations

Granting Access to the Scheduler

- Requires use of SURROGAT Profiles
  - Grants job submitter permission to submit a job with a specified USERID and no password
  - Once submitted, the job runs under the authority of the USERID

Defining SURROGAT Profiles

```
RDEF SURROGAT AAA#SP00.SUBMIT UACC(NONE) OWNER(AAA)
RDEF SURROGAT AAA#SP01.SUBMIT UACC(NONE) OWNER(AAA)
RDEF SURROGAT AAA#SP02.SUBMIT UACC(NONE) OWNER(AAA)
RDEF SURROGAT AAA#SP99.SUBMIT UACC(NONE) OWNER(AAA)
RDEF SURROGAT BBB#SP00.SUBMIT UACC(NONE) OWNER(BBB)
RDEF SURROGAT CCC#SP00.SUBMIT UACC(NONE) OWNER(CCC)
```

Notes:

- Define USERID.SUBMIT SURROGAT profile for each SPJ id
- Universal Access (UACC) should always be NONE!

RACF Security Recommendations

Granting Access to the Scheduler

Permitting access to SURROGAT profiles

```
PE AAA#SP00.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ)
PE AAA#SP01.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ)
PE AAA#SP02.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ)
PE AAA#SP99.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ)
PE BBB#SP00.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ
PE CCC#SP00.SUBMIT CLASS(SURROGAT) ID(scheduler-id) ACCESS(READ)
```

Notes:

- The USERID associated with the scheduler MUST be granted READ access to every SPJ SURROGAT profile
- Access need never be higher than READ
- ALTER access should be avoided

Monitoring

Ensure appropriate reports are produced regularly and automatically to facilitate auditing

RACF reporting can provide:

- **Details on what the SPECjob did by showing:**
  - **USERID requesting access**
  - **All files accessed**
  - **Access level requested (i.e. Read, Update, Control, Alter)**
  - **Access level granted**
  - **Jobname**
- **Details on the use of EMERGENCY packages (depending upon the BC1TNEQU configuration within Endevor's ESI) by showing:**
  - **Package creator (i.e. user requesting access to package profiles)**
  - **ESI Profiles, protected resources accessed and the level of access**
  - **Package Approvers**

Monitoring

Endevor reporting can provide:

- **Details on each action performed against SPJ type elements within a specified Endevor inventory (CONRPT42) including USERID, date, VV.LL**
- **Package summary and package approver information (CONRPT71) by Package Id including quorum specifications and the ids associated with the person(s) who performed the approval(s)**

Summary

Ad-hoc production file updates are necessary and typically not performed using a standardized, preauthorized process

In particular, emergency (Firecall) situations are prone to creating exposures and greatly increasing risk

Simply by the nature of its design, Endevor can facilitate an automated process which eliminates exposures and significantly reduces risk

Configuration involves:

- **Exploiting JCL validation software tools**
- **Coding processors that interface with JCL validation software and scheduling software**
- **Designing and configuring security such that special ids have the necessary access**
- **Providing reports that ensure appropriate monitoring is performed**

Questions