Integrating RTC Enterprise with ChangeMan Package Management and Build

Scott Pecnik
PacGenesis, Inc.

August 7, 2014
Room 315

www.PacGenesis.com

#SHAREorg

SHARE in Pittsburgh 2014

Then and Now

- Then
  - Process driven
  - Proven technology
  - Substantial financial investment
  - Relatively stable and bug free

- Now
  - Agile Software Development Methodology
  - Traceability
  - Modern devices, Android/iPad
  - Web 2.0

Bridging the Gap

- Two mainframe Developers united by IBM Rational Software
- They are the Now
- What are the Challenges?
- Cultural Differences
- Technology Driven vs. Process Driven

Getting There

- Rational Software is the entry point
- A way for the “new” to play with the old
- A way for the “old” to play with the new
- Comprehensive adoption plan
- Business investment

RTC/RDz - *What are they?*

Case Study: Worldwide rollout of RTC at a Large Financial Institution

Why

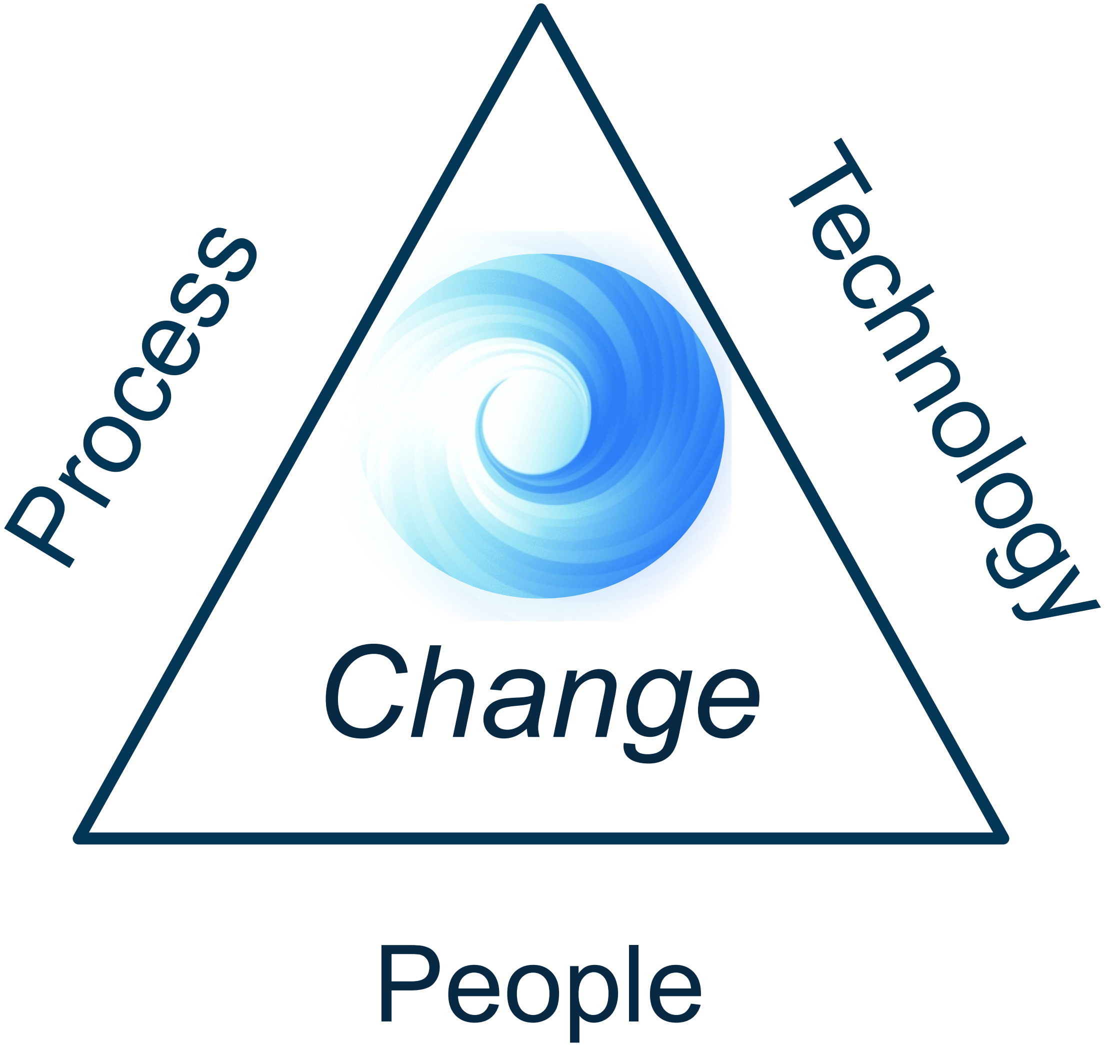
From…

- Limited transparency into resource usage, quality and project delivery
- Non-integrated development toolset, requiring developers to jump between tools
- Relatively rigid waterfall approach to developing software
- Limited adoption of modern development practices, such as Continuous Integration

To…

- Improved management, visibility, and quality assurance of projects and teams (including offshore resources), with better estimating, tracking, and reporting
- Integrated toolset through single window interface across functional areas (build, test, defect management)
- Support for Agile and hybrid SDLC methodologies in addition to traditional approaches
- Deployment of continuous integration capabilities

How



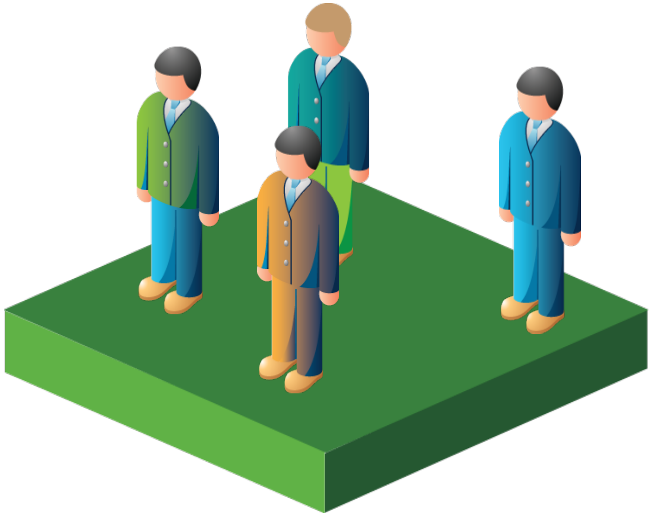
Picking the Team

- Executive Sponsorship
- Project Management
- Development Team

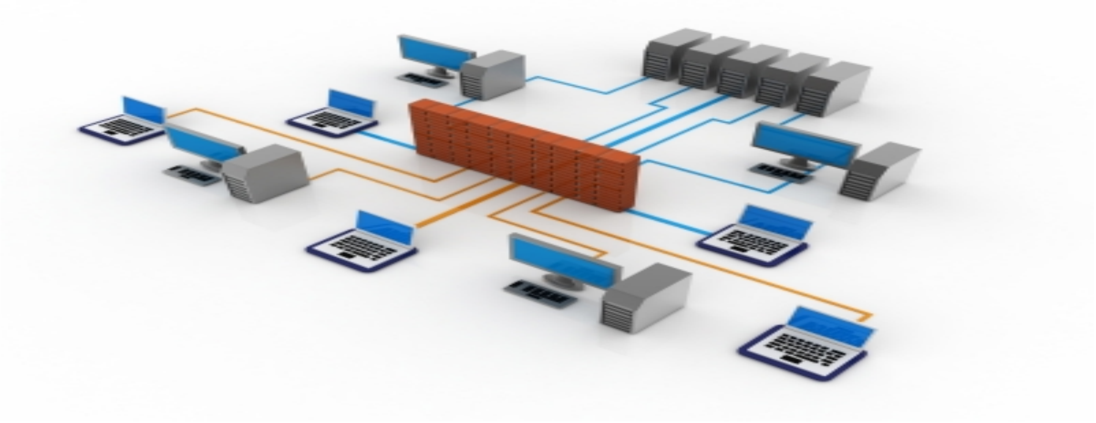
→ Leadership

→ Enthusiasts

- Infrastructure Team

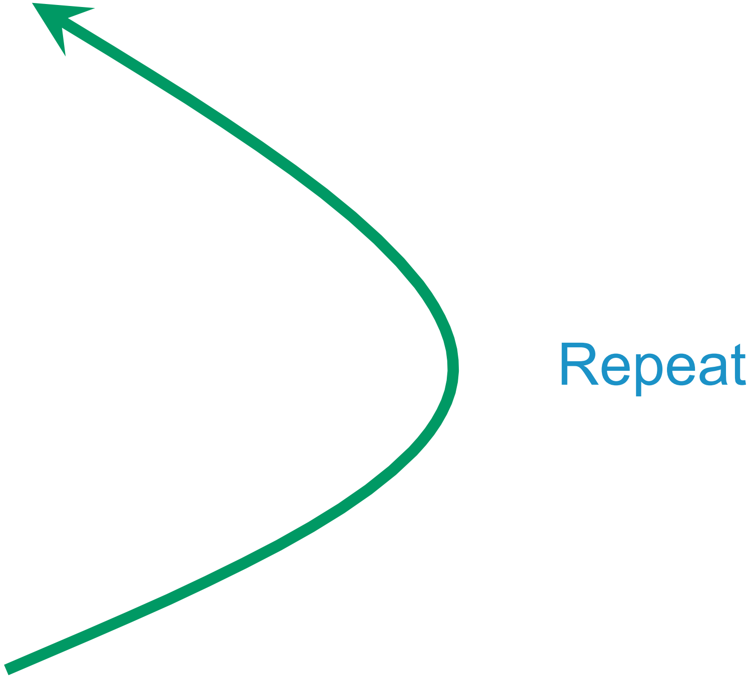


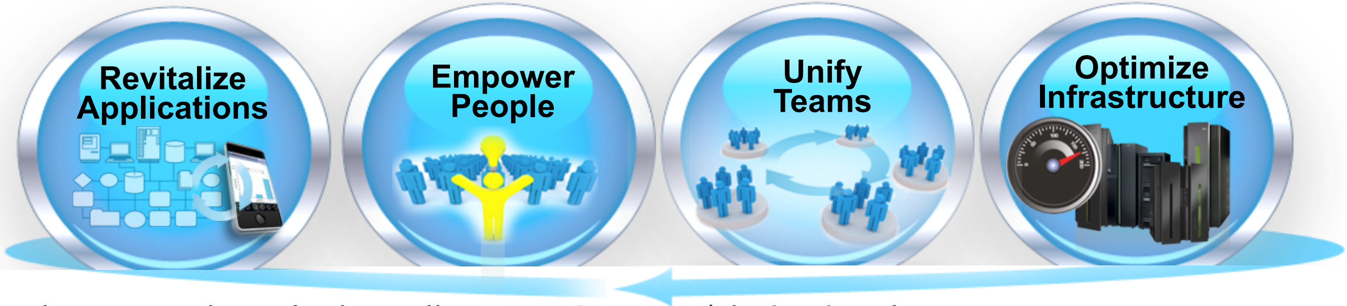
Modern, distributed tooling requires a range of skills and expertise to install, configure and optimize software



Tooling and Process Adoption

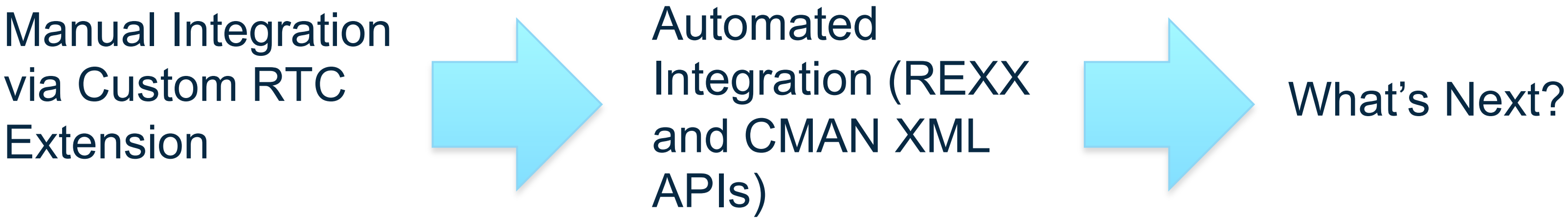
- Pick a Team
  They serve as evangelists moving forward
- … And a Project
  Manageable yet strategic to the business
- Identify Goals
  Need to measure success
- Deploy the Solution
  Use it in production
- Capture Best Practices
  What worked, what didn't, reuse





What

From RTC to Existing Build and Deploy System



Manual Integration via Custom RTC Extension

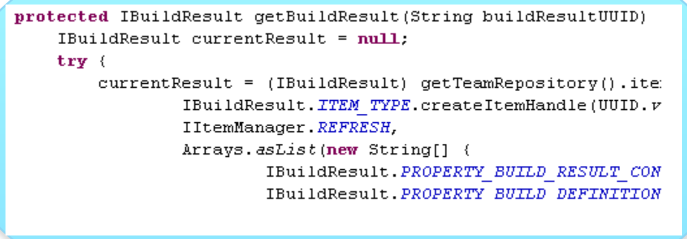
Automated Integration (REXX and CMAN XML APIs)

What’s Next?

What

Manual Integration via Custom RTC Extension

Developer Manually Stages into CMAN

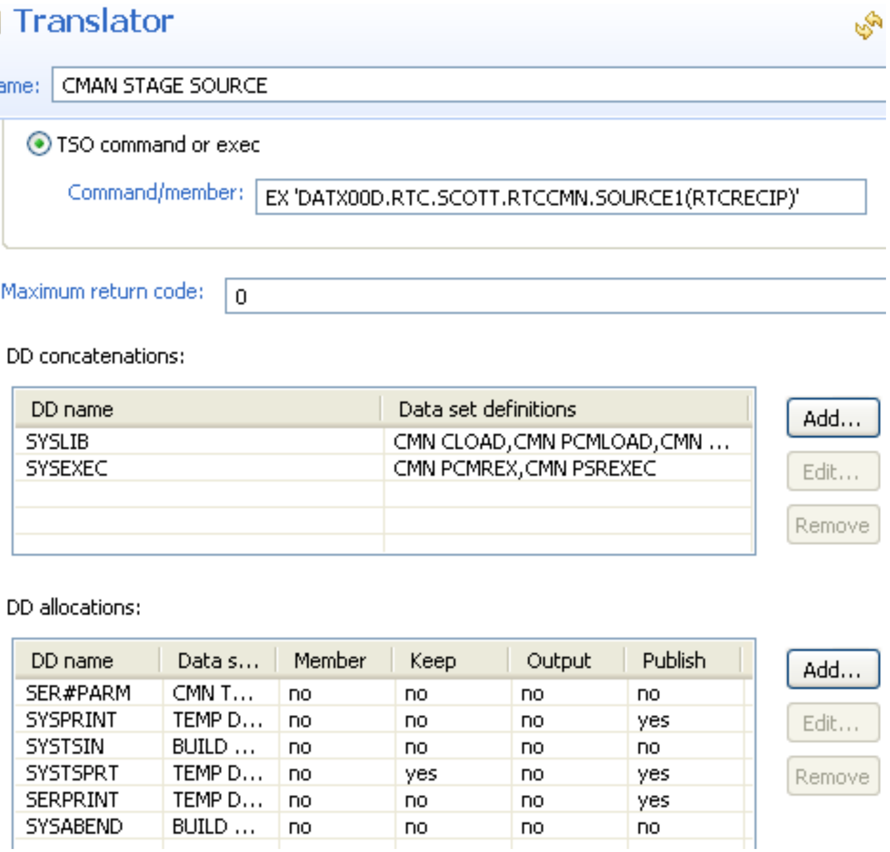


```
protected IBuildResult getBuildResult(String buildResultUUID)
    IBuildResult currentResult = null;
    try {
        currentResult = (IBuildResult) getTeamRepository().ite
                IBuildResult.ITEM_TYPE.createItemHandle(UUID.v
                IItemManager.REFRESH,
                Arrays.asList(new String[] {
                        IBuildResult.PROPERTY_BUILD_RESULT_CON
                        IBuildResult.PROPERTY_BUILD_DEFINITION
```

What

Automated Integration (REXX and CMAN XML APIs)

No Developer Impact



Translator

ame: CMAN STAGE SOURCE

● TSO command or exec

Command/member: EX 'DATX00D.RTC.SCOTT.RTCCMN.SOURCE1(RTCRECIP)'

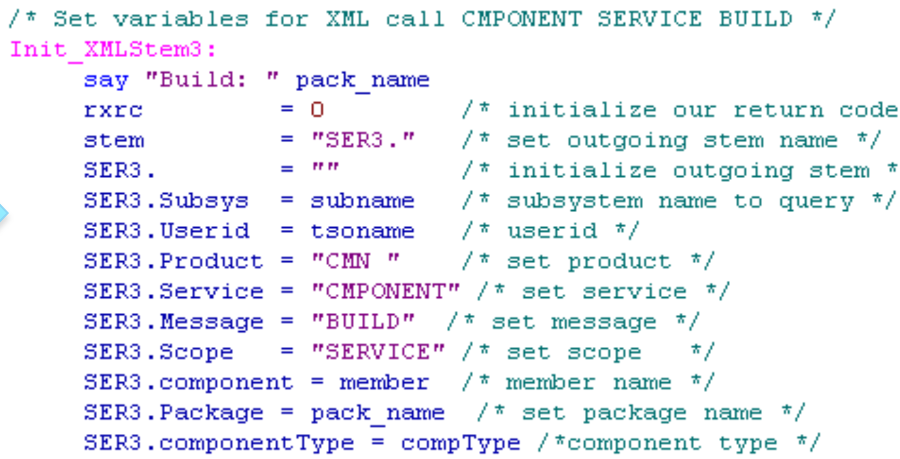
Maximum return code: 0

DD concatenations:

DD name	Data set definitions
SYSLIB	CMN CLOAD,CMN PCMLOAD,CMN ...
SYSEXEC	CMN PCMREX,CMN PSREXEC

DD allocations:

DD name	Data s...	Member	Keep	Output	Publish
SER#PARM	CMN T...	no	no	no	no
SYSPRINT	TEMP D...	no	no	no	yes
SYSTSIN	BUILD ...	no	no	no	no
SYSTSPRT	TEMP D...	no	yes	no	yes
SERPRINT	TEMP D...	no	no	no	yes
SYSABEND	BUILD ...	no	no	no	no

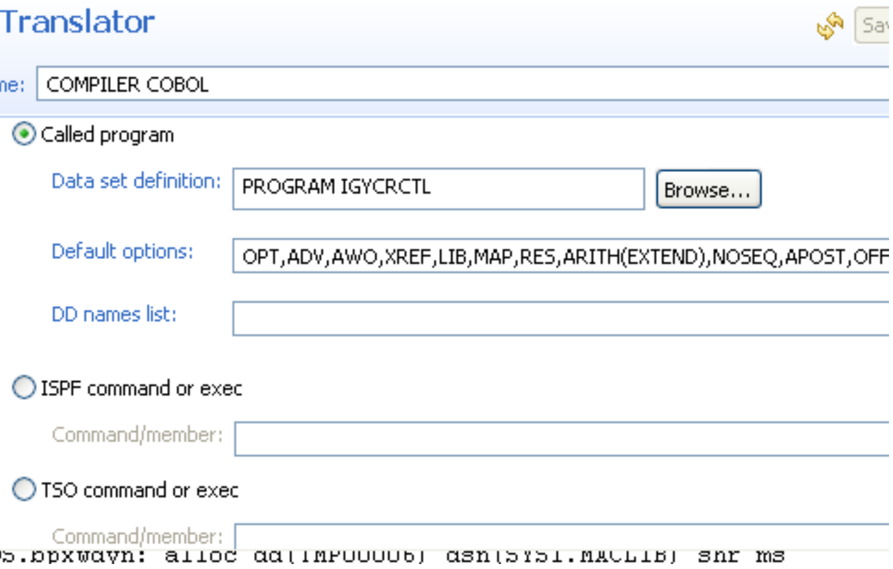


```
/* Set variables for XML call CMPONENT SERVICE BUILD */
Init_XMLStem3:
     say "Build: " pack_name
     rxrc           = 0          /* initialize our return code
     stem           = "SER3."    /* set outgoing stem name */
     SER3.          = ""         /* initialize outgoing stem *
     SER3.Subsys    = subname    /* subsystem name to query */
     SER3.Userid    = tsoname    /* userid */
     SER3.Product = "CMN "       /* set product */
     SER3.Service = "CMPONENT" /* set service */
     SER3.Message = "BUILD"    /* set message */
     SER3.Scope     = "SERVICE" /* set scope    */
     SER3.component = member    /* member name */
     SER3.Package = pack_name    /* set package name */
     SER3.componentType = compType /*component type */
```

What

What’s Next?

Native RTC Build/Compile



Translator

me: COMPILER COBOL

● Called program

Data set definition: PROGRAM IGYCRCTL

Default options: OPT,ADV,AWO,XREF,LIB,MAP,RES,ARITH(EXTEND),NOSEQ,APOST,OFF

DD names list:

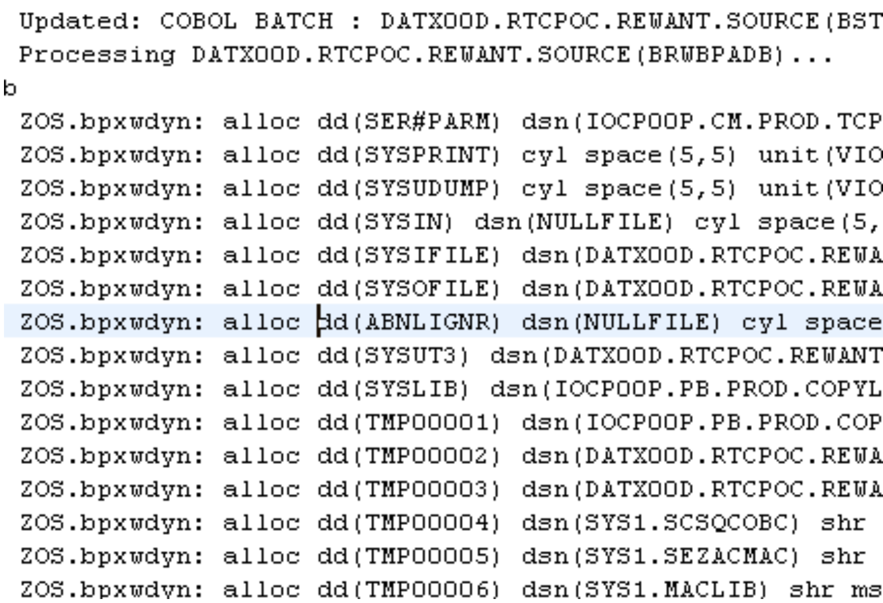
○ ISPF command or exec

Command/member:

○ TSO command or exec

Command/member:

S.bpxwdyn: alloc dd(TMP00006) dsn(SYS1.MACLIB) shr ms



```
 Updated: COBOL BATCH : DATXOOD.RTCPOC.REWANT.SOURCE(BST
 Processing DATXOOD.RTCPOC.REWANT.SOURCE(BRWBPADB)...
b
 ZOS.bpxwdyn: alloc dd(SER#PARM) dsn(IOCPOOP.CM.PROD.TCP
 ZOS.bpxwdyn: alloc dd(SYSPRINT) cyl space(5,5) unit(VIO
 ZOS.bpxwdyn: alloc dd(SYSUDUMP) cyl space(5,5) unit(VIO
 ZOS.bpxwdyn: alloc dd(SYSIN) dsn(NULLFILE) cyl space(5,
 ZOS.bpxwdyn: alloc dd(SYSIFILE) dsn(DATXOOD.RTCPOC.REWA
 ZOS.bpxwdyn: alloc dd(SYSOFILE) dsn(DATXOOD.RTCPOC.REWA
 ZOS.bpxwdyn: alloc dd(ABNLIGNR) dsn(NULLFILE) cyl space
 ZOS.bpxwdyn: alloc dd(SYSUT3) dsn(DATXOOD.RTCPOC.REWANT
 ZOS.bpxwdyn: alloc dd(SYSLIB) dsn(IOCPOOP.PB.PROD.COPYL
 ZOS.bpxwdyn: alloc dd(TMP00001) dsn(IOCPOOP.PB.PROD.COP
 ZOS.bpxwdyn: alloc dd(TMP00002) dsn(DATXOOD.RTCPOC.REWA
 ZOS.bpxwdyn: alloc dd(TMP00003) dsn(DATXOOD.RTCPOC.REWA
 ZOS.bpxwdyn: alloc dd(TMP00004) dsn(SYS1.SCSQCOBC) shr
 ZOS.bpxwdyn: alloc dd(TMP00005) dsn(SYS1.SEZACMAC) shr
 ZOS.bpxwdyn: alloc dd(TMP00006) dsn(SYS1.MACLIB) shr ms
```

RTC – *Compiling and Integrating*

Questions

Contact:

Scott Pecnik, PacGenesis
specnik@pacgenesis