



IBM i & git

Pinaki Basu

Omni User Group - October 2020

Abstract

Open Source solutions on IBM i, enables teams to modernize their application development space.

In this session we will take a look at how we can introduce “git” into your workflow without changing existing IBM i Change Management workflow.

Start sending all your changes to a Git repository, and using that repository for change analysis and doing code review.

In this session will setup a “git” repository on your IBMi and add objects to it.

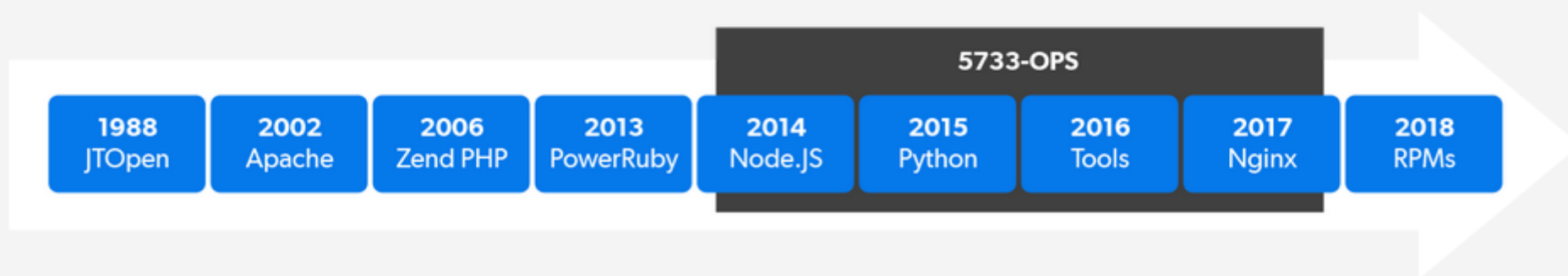
Agenda

1. OpenSource on IBMi
2. What is Git
3. Current Change Management Tools on IBMi
4. Integrate Git into your current workflow
5. How to use Git for Code Analysis / Review
6. Demo
7. Q & A

Open Source and IBM i

IBM has a long history of both contributing to as well as adopting various open source standards like OpenSSL and TLS as well as open source solutions.

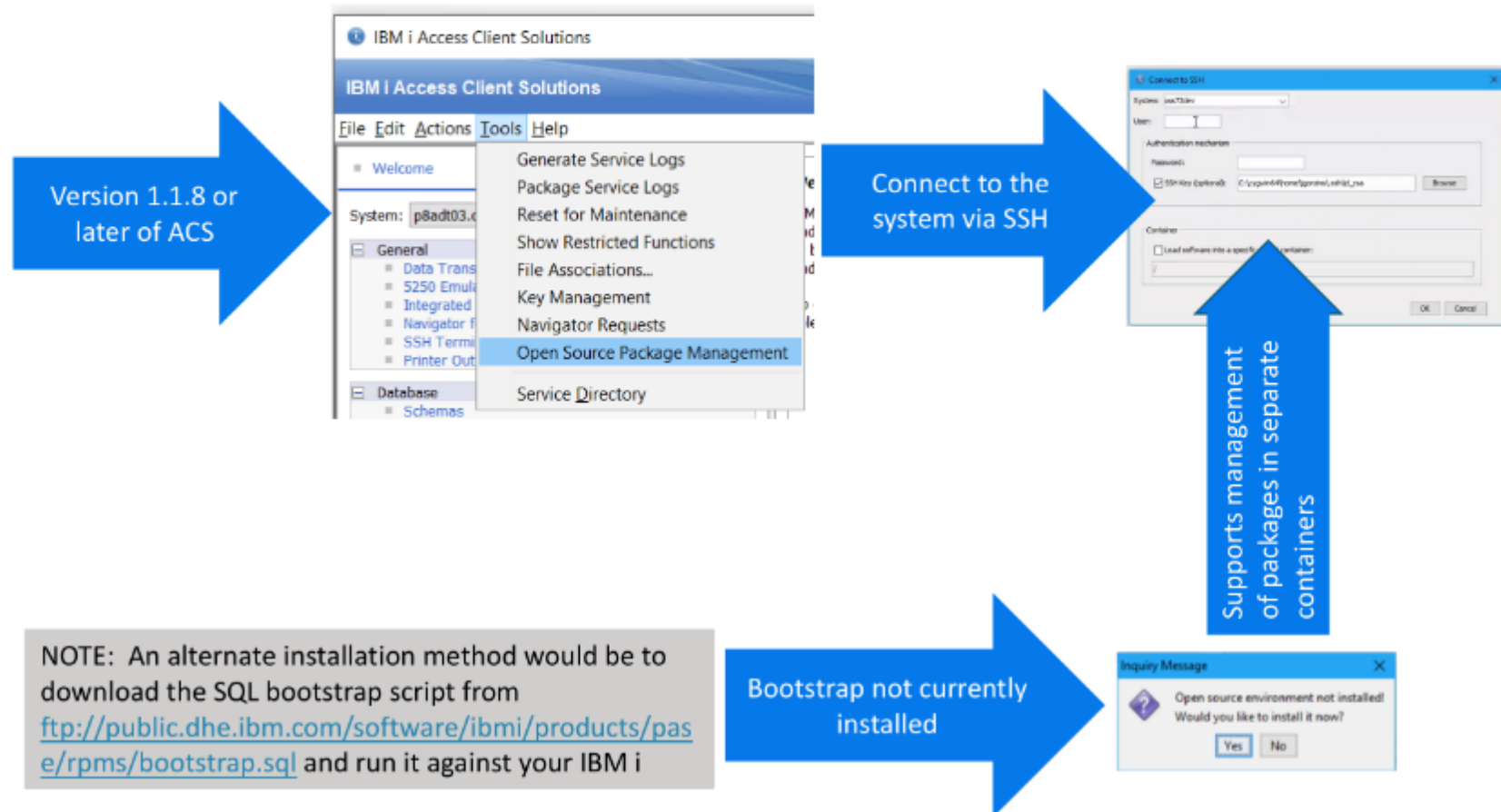
Consider the following timeline:



July 2018+

Bootstrapping OSS

Bootstrapping is the process of installing utilities and repository definitions to enable the system with the necessary commands for managing open source packages



NOTE: An alternate installation method would be to download the SQL bootstrap script from <ftp://public.dhe.ibm.com/software/ibmi/products/pase/rpms/bootstrap.sql> and run it against your IBM i

Bash Shell - SSH

```
bash-4.4$ ls -l
total 1336
drwxr-sr-x  2 pbasu  0          8192 Apr 11 2020 QCLLESRC
drwxr-sr-x  2 pbasu  0       180224 Apr 11 2020 QCLSRC
drwxr-sr-x  2 pbasu  0          8192 Apr 11 2020 QCMDSRC
drwxr-sr-x  2 pbasu  0     229376 Apr 11 2020 QDDSSRC
drwxr-sr-x  2 pbasu  0          8192 Apr 11 2020 QPNLSRC
drwxr-sr-x  2 pbasu  0     188416 Apr 11 2020 QRPGLSRC
drwxr-sr-x  2 pbasu  0         24576 Apr 11 2020 QRPGRSRC
-rw-rw-rw-  1 pbasu  0           24 Aug 13 16:04 curl_log.txt
drwxr-sr-x  2 pbasu  0          8192 May 13 12:18 downloads
-rw-rw-rw-  1 pbasu  0           24 Aug 13 16:03 imports
drwxr-sr-x  2 pbasu  0         12288 Aug 13 15:49 scripts
bash-4.4$ uname -a
OS400 [REDACTED] 3 7 [REDACTED]
bash-4.4$
```

Putty.exe*[64]:25844 ◀ 191012[64] 5/5 [+] NUM InpGrp PRI 135x29

Bash Shell

```
bash-4.4$ ps -ef
  UID  PID  PPID  C   STIME  TTY  TIME  CMD
  qsecofr  11   10   0   Oct 10   -   0:30  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 465
  qsecofr  33    1   0   Oct 10   -   0:47  [cimserve]
 qypsjsvr  49    1   0   Oct 10   -   0:35  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 584
  qlwisvr  64    1   0   Oct 10   -   0:51  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 963
4294770734  65    1   0   Oct 10   -   1:00  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 985
  qlwisvr  66    1   0   Oct 10   -   7:12  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 1397
  qlwisvr  67    1   0   Oct 10   -   2:09  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 979
  qlwisvr  68    1   0   Oct 10   -   0:50  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 979
   qsys   77    1   0   Oct 10   -   1:17  /QIBM/ProdData/OS/SLP/bin/lslp-kernel -f/QIBM/UserData/OS/SLP/conf/ibmslp.conf
  qsrvagt  79    1   0   Oct 10   -   0:29  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 834
  qlwisvr  93   92   0   Oct 10   -   1:22  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 470
  qejbsvr 381    1   0   Oct 10   -  12:26  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 27952
  qsecofr 2383   1   0   Oct 12   -   0:00  /QOpenSys/QIBM/ProdData/SC1/OpenSSH/sbin/sshd
  qejbsvr 4845   1   0   Oct 13   -   8:49  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 27961
oneworld 7925 7913  0   Oct 17   -   0:25  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 1034
oneworld 7927 7916  0   Oct 17   -   2:37  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 1030
   252  7948 7932  0   Oct 17   -   0:22  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 9494
   252  7957 7932  0   Oct 17   -   1:04  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 2136
oneworld 7959 7933  0   Oct 17   -   1:01  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 9402
oneworld 7961 7933  0   Oct 17   -   1:02  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 9369
  qejbsvr 7966   1   0   Oct 17   -   6:17  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 104926
oneworld 7968 7933  0   Oct 17   -   1:18  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/ppc/jvmStartPase 1972
  qejbsvr 8002   1   0   Oct 17   -   9:42  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 104357
  qejbsvr 8030   1   0   Oct 17   -   3:02  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 110214
  qejbsvr 8058   1   0   Oct 17   -   5:52  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 109675
  qejbsvr 8108   1   0   Oct 17   -   7:10  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 109661
  qejbsvr 8142   1   0   Oct 17   -   4:28  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 110052
  qejbsvr 8171   1   0   Oct 17   -   6:31  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 110032
  qejbsvr 8200   1   0   Oct 17   -   3:56  /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 109343

Putty.exe*[64]:25844                                     « 191012[64] 5/5 [+] NUM InpGrp PRI 135x29 (1,7) 25V
```



git

What is Git

What Is Git?

Git is a distributed, open-source [version control](#) system (VCS) that enables you to store code, track revision history, merge code changes, and revert to earlier code version when needed.

What Are the Benefits of Git?

Historical Change Tracking

Work as a Team

Improve Team Speed & Productivity


Availability and Redundancy

Git Is the Industry Standard

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

 **git** --distributed-even-if-your-workflow-isnt

 **git** --everything-is-local

 **git** --distributed-is-the-new-centralized

 **git** --fast-version-control

Introduced in April 2005. Released to manage the Linux Kernel

Name	2015	2017	2018
Git	69.3%	69.2%	87.2%
Subversion	36.9%	9.1%	16.1%
TFVC	12.2%	7.3%	10.9%
Mercurial	7.9%	1.9%	3.6%
CVS	4.2%	0%	0%
Perforce	3.3%	0%	0%
VSS	0%	0.6%	0%
ClearCase	0%	0.4%	0%
Zip file backups	0%	2.0%	7.9%
Raw network sharing	0%	1.7%	7.9%
Other	5.8%	3.0%	0%
None	9.3%	4.8%	4.8%





BIG 3



TURNOVER V100



IMPLEMENTER

Other Change Management Tools

1. ARCAD-Skipper
2. SEE/CHANGE
3. Change Master
4. IBM Rational ClearCase
5. TAATOOLS – SRCCTL



Integrate Git into your process

Install Git

1. Install OpenSource on your IBMi
2. Use ACS or YUM install git
3. Create a Source Code Directory on your IFS - using SSH Shell or QSH
4. Initialize a git repository on the Source Code Directory on the IFS

```
*****START - GIT Initialize & Setting up the Remote Repository*****
```

```
git init
```

```
git add
```

```
git commit
```

```
git remote add AS400_Source http://remotegitserver/git/AS400_Source.git
```

```
git push --set-upstream AS400_Source master
```

```
git remote -v
```

```
AS400_Source http://wdirs1/git/AS400_Source.git (fetch)
```

```
AS400_Source http://wdirs1/git/AS400_Source.git (push)
```

```
*****END - GIT Initialize & Setting up the Remote Repository*****
```

Integrate Git into your process

Create Source Copy Process

1. Create a CL program to copy Source from your “Final” Source code PF – based on date changed of “source member”

```
CPYTOIMPF FROMFILE(&MLLIB/&MLFILE &MLNAME) +  
          TOSTMF('\0_AS400_Source_Code\' |< &LIB |< +  
          '\ ' |< &FIL |< '\ ' *CAT &MLNAME |< '.' |< +  
          &MLSEU2) MBROPT(*REPLACE) +  
          STMFCCSID(*PCASCII) RCDDL(*CRLF) +  
          DTAFMT(*FIXED)
```

2. Run the “git add .” command in QSH
3. Run the “git commit –m ‘Changes for xxxx date’ ” command in QSH

RPG Driver Program

```
CTL-OPT;
*****
*
*           Maurice Sporting Goods
*
*****
* SR.....: Sync Sources
* DATE.....: 06/22/2018
* PROGRAMMER.....: Mark Kasper
* FUNCTION.....: Sync source to IFS based on Database.
*****
*           MAINTENANCE HISTORY
*****
* Mod Sr#      Date      Pgmr      Modification
*-----
/001 * 001        06/27/18  MPK        Have an option to sync all the
/001 *                               sources.
*****

dcl-f table1   disk USAGE(*INPUT) Keyed;

dcl-pr SYNCSRC01;
PCHANGES CHAR(1);
End-PR;

dcl-pi SYNCSRC01;
PCHANGES CHAR(1);
End-Pi;

// Define Procedures

dcl-pr CLSYNCSRC EXTPGM;
PLibrary CHAR(10);
PFILE Char(10);
PALL Char(1);
END-PR;

// Define Various Fields

dcl-s kyTb1Sou char(8);
dcl-s kyTb1Fld char(6);
dcl-s kyTb1A Char(5);
dcl-s kyTb1N Zoned(5);
dcl-s PLibrary Char(10);
dcl-s PFile Char(10);
dcl-s PAll Char(1);
*****
**                               Main Program                               **
*****

kyTb1Sou = 'SYNC';
kyTb1Fld = 'SOURCE';
Setll (kyTb1Sou:kyTb1Fld) table1;
reade (kyTb1Sou:kyTb1Fld) table1;
Dow not %EOF(table1);
  If TB1FL = 'Y';
    Clear PLibrary;
    Clear PFile;
    Clear PALL;
    PLibrary = %Subst(TB1DES:1:10);
    PFile = TB1DE2;
    PALL = TB1F2;
/001 //001 CallP CLSYNCSRC(PLibrary:PFile);
/001 CallP CLSYNCSRC(PLibrary:PFile:PALL);
  ENDIF;
reade (kyTb1Sou:kyTb1Fld) table1;
ENDDO;
*INLR = *On;
```

CL Copy to IFS - source Code

```

/*****
/*****      Maurice Sporting Goods
/*****
/*****      Program name: CLSYNCSRC- Send source Program to IFS.
/*****      Written by : Mark Kasper
/*****      Date Written: June 14, 2018
/*****
/*****      Modification History
/*****      Mod Task      Date      Pgm      Description
/*****      -----      -
/*****      001          06/18/2018  MPK      Change the CPYTOIMPF to use the
/*****      field call &MLSEU2, instead of
/*****      &MLSEU. The new column has a
/*****      better description of the file
/*****      type. Also changed the logic to
/*****      only run the GIT statement when
/*****      there a source that was changed.
/*****      002          06/21/2018  MPK      If the time is less than or equal
/*****      to 2am, then use the previous day.
/*****      003          06/22/2018  MPK      If the folder does not exist, go
/*****      ahead and create the AS400 Folder.
/*****      004          06/25/2018  MPK      Replace MLCDAT (Created Date) with
/*****      MLCHGD (Changed Date). This will
/*****      ensure that all the changes are
/*****      copied - vs only new source code.
/*****      005          06/27/2018  MPK      Change the CPYTOIMPF command to
/*****      use the *REPLACE, instead of *ADD.
/*****      Add additional parameter to send
/*****      all the source.
/*****
PGM          PARM(&LIB &FIL &ALL)

DCL          VAR(&ALL)      TYPE(*CHAR)  LEN(1)
DCL          VAR(&CMD)      TYPE(*CHAR)  LEN(100)
DCL          VAR(&LIB)      TYPE(*CHAR)  LEN(10)
DCL          VAR(&FIL)      TYPE(*CHAR)  LEN(10)
DCL          VAR(&CONT)     TYPE(*CHAR)  LEN(01)

DCL          VAR(&JOBDATE)  TYPE(*CHAR)  LEN(6)
DCL          VAR(&JOBDATE2) TYPE(*DEC)  LEN(6 0)
DCL          VAR(&JOBDATE3) TYPE(*DEC)  LEN(8 0)
DCL          VAR(&JOBDATE4) TYPE(*CHAR)  LEN(8)
DCL          VAR(&DATE1)   TYPE(*DEC)  LEN(8 0)

DCL          VAR(&DATE)     TYPE(*CHAR)  LEN(8)
DCL          VAR(&QTE)      TYPE(*CHAR)  LEN(1) VALUE('')
DCL          VAR(&SYSHR)    TYPE(*CHAR)  LEN(2)
DCL          VAR(&PATH)     TYPE(*CHAR)  LEN(100)

DCLF         FILE(MBRLIST2)

CHGVAR       VAR(&PATH) VALUE('/0_AS400_Source_Code/' |< +
&LIB)-

CHGCURDIR   DIR(&PATH)
MONMSG      MSGID(CPFA0A9) EXEC(DO)
MKDIR       DIR(&PATH)
ENDDO

CHGVAR       VAR(&PATH) VALUE('/0_AS400_Source_Code/' |< +
&LIB |< '/' |< &FIL)-

CHGCURDIR   DIR(&PATH)
MONMSG      MSGID(CPFA0A9) EXEC(DO)
MKDIR       DIR(&PATH)
ENDDO

```

```

RTVSYSVAL   SYSVAL(QHOUR) RTNVAR(&SYSHR)
RTVJOBA     DATE(&JOBDATE)
CHGVAR      VAR(&JOBDATE2) VALUE(&JOBDATE)

CALL        PGM(SWR001) PARM(&JOBDATE2 &JOBDATE3)

CHGVAR      VAR(&JOBDATE4) VALUE(&JOBDATE3)

If          (&SYSHR *LT '02') then(DO)
CALL       PGM(SWC014) PARM(&JOBDATE4 '0001' '-' 'D')
ENDDO

DSPFD      FILE(&LIB/&FIL) OUTPUT(*OUTFILE) +
           OUTPUTFILE(QTEMP/MBRLIST2)

CHGVAR      VAR(&CONT) VALUE('Y')

RCVF       MSGID(CPF0864) EXEC(CHGVAR VAR(&CONT) +
VALUE('N'))

DOWHILE    COND(&CONT = 'Y')

CHGVAR     VAR(&DATE1) VALUE(&MLCHGD)
If         (&MLCCEN = '0') then(DO)
CHGVAR     VAR(&DATE1) VALUE(&DATE1 + 19000000)
ENDDO
ELSE      CMD(DO)
CHGVAR     VAR(&DATE1) VALUE(&DATE1 + 20000000)
enddo

CHGVAR     VAR(&DATE) VALUE(&DATE1)
IF         COND(&DATE = &JOBDATE4 *AND &ALL = 'N' *OR +
&ALL = 'Y') THEN(DO)

CPYTOIMPF  FROMFILE(&MLLIB/&MLFILE &MLNAME) +
           TOSTMF('/0_AS400_Source_Code/' |< &LIB |< +
           '/' |< &FIL |< '/' *CAT-&MLNAME |< '.' |< +
           &MLSEU2) MBROPT(*REPLACE) +
           STMFCCSID(*PCASCII) RCDDL(*CRLF) +
           DTAFMT(*FIXED)

ENDDO
RCVF       MSGID(CPF0864) EXEC(CHGVAR VAR(&CONT) +
VALUE('N'))
ENDDO

CHGVAR     VAR(&CMD) VALUE('cd /0_AS400_Source_Code && git +
add .' )
STRQSH     CMD(&CMD)

CHGVAR     VAR(&CMD) VALUE('cd /0_AS400_Source_Code && +
git commit -m ' |< &QTE |< 'Changes for ' +
|< &JOBDATE4 |< &QTE)
STRQSH     CMD(&CMD)

ENDPGM:    ENDPGM

```


Why did we do it this way

1. Do not disrupt existing checkin / checkout process for code – Easy to use
2. New developers / New projects / Multiple Source Code libraries
3. Rapid changes to 35+ Year code base, RPGII, RPGIII, RPGIV and RPG Free
4. Started using RDi / SEU limitations
5. Developers were responsible for pushing code to production
6. Needed ability to review change sets in block
7. Needed ability to view Before & After vs – “change #” or “change comments”
8. Needed ability to access Source Code across all Source Code libraries

Code Review before git



Developers

DO

1. Code review
2. Quick review of Code changes
3. Ability to drill down into Code sets using a browser
4. Increased productivity

NOT DO

1. No change to development workflow
2. Did not use git command line
3. No training

DEMO

Demo

&

Q & A