Quickly Create Powerful IBM i System Commands with Java and CL

Presenter

Richard Schoen
Director of Document Management
26+ years of multi-platform development

Agenda

- **Why use Java and CL Together**
- Review development environments for Java
- Code, compile and build
- Work through sample commands

No Hamburger Flipping

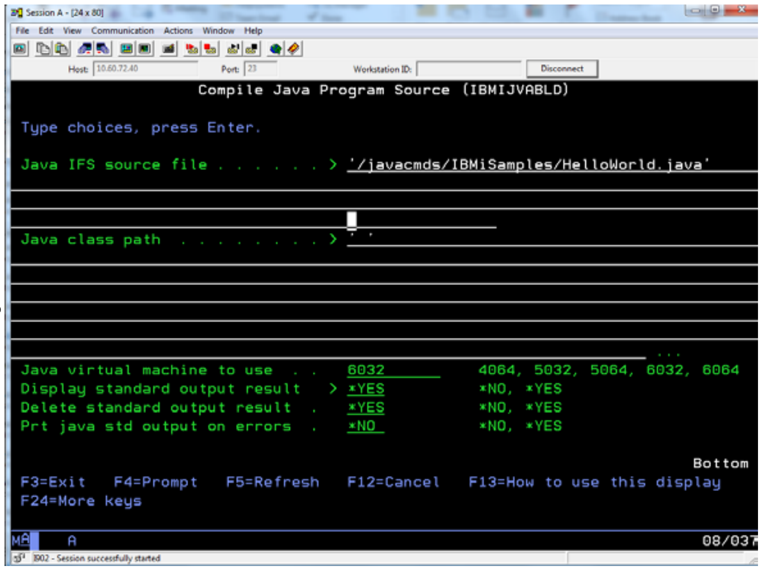
Why Use Java with CL ?

- Functionality not easily available in CL or RPG by themselves
- JT400 for IBM i local and remote service access

  DB, Commands, Program Calls, Data Queues, Etc.
- Access to network shares (send and receive files via Netbios)
- Read and modify XML, XLS and Text files
- PDF manipulation – iText. (create, read, write, merge, etc.)
- Excel manipulation – Apache POI
- Submit jobs to remote systems and return data from the calls
- My samples help make Java accessible from 5250 environment

What are CL Commands ?

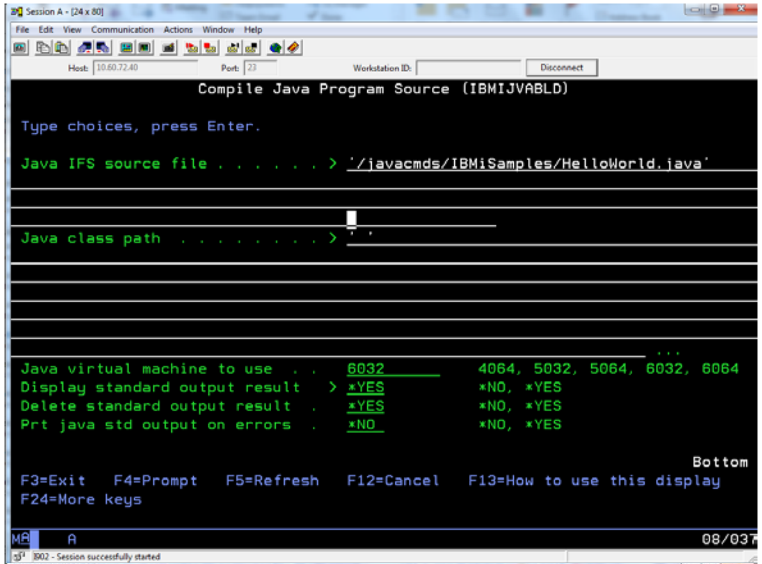
- CL commands front end calls to CL, RPG or COBOL programs.
- Takes prompted input parameters.
- Can return parameters.
- Can be embedded in other CL programs.
- Entered data values are padded to parm length.



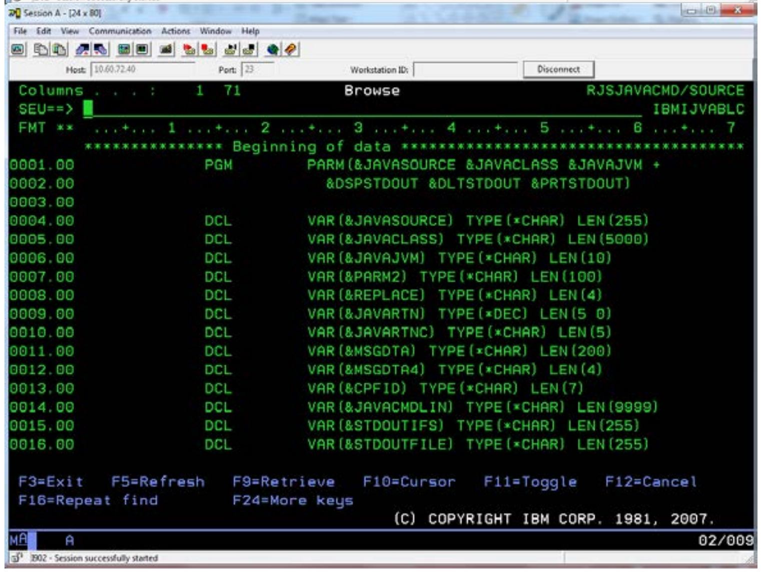
```
                  Compile Java Program Source (IBMIJVABLD)

 Type choices, press Enter.

 Java IFS source file . . . . . . > '/javacmds/IBMiSamples/HelloWorld.java'



 Java class path  . . . . . . . . > ' '

                                                                        ...
 Java virtual machine to use  . .   6032          4064, 5032, 5064, 6032, 6064
 Display standard output result   > *YES          *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
 Prt java std output on errors  .   *NO           *NO, *YES

                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     A                                                                 08/037
I902 - Session successfully started
```

- Simple or complex parameters lists can be passed to a command.
- Parameter valued can be validated.
- Validity checking and prompt override programs.

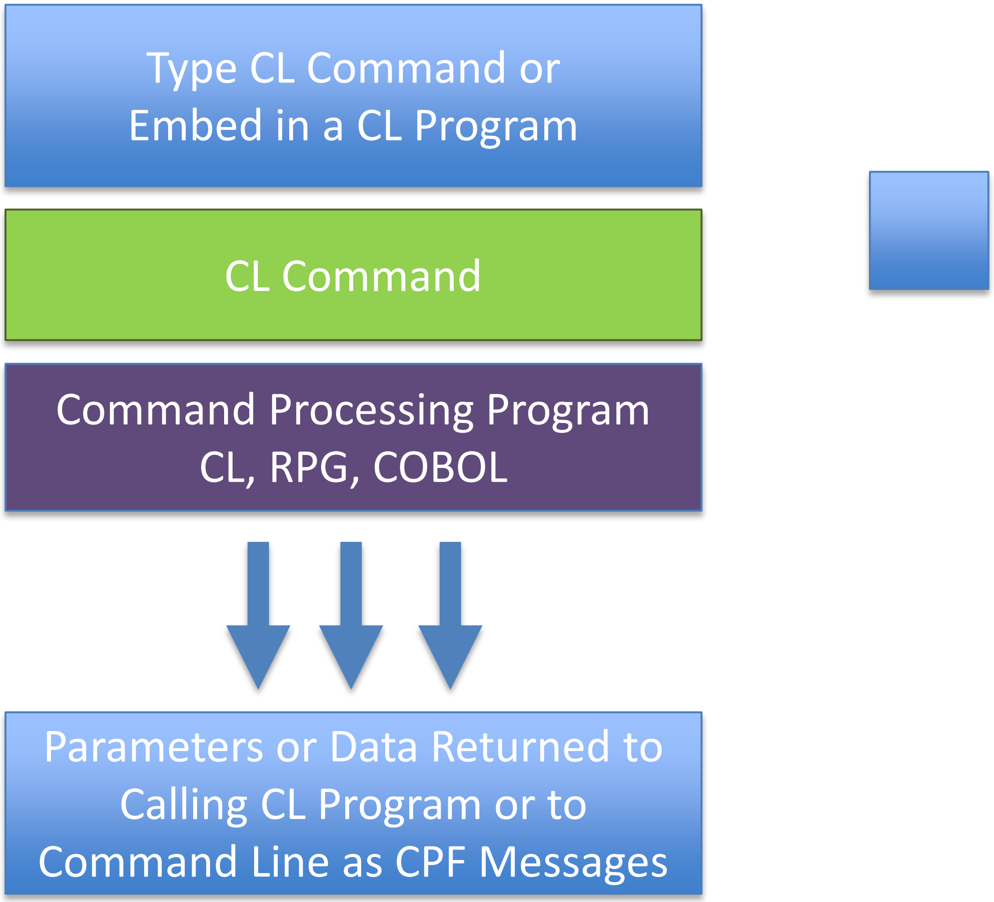
Traditional CL Command Process Structure



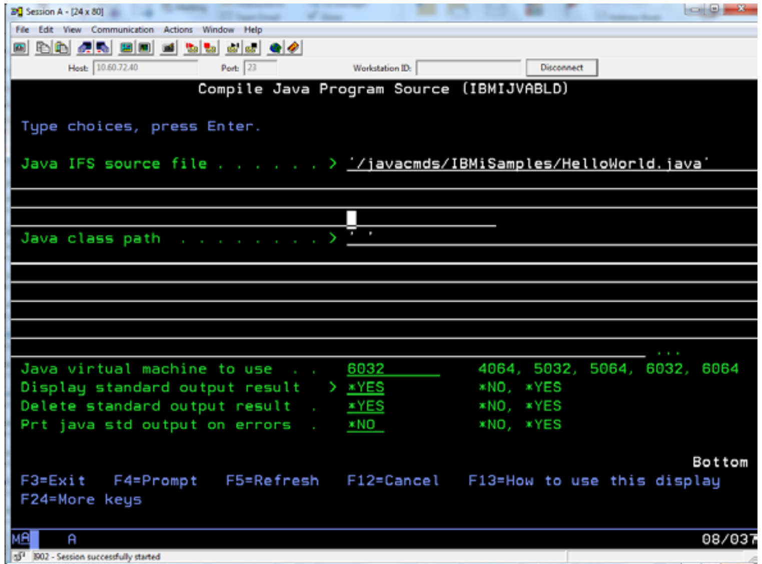
```
                     Compile Java Program Source (IBMIJVABLD)

 Type choices, press Enter.

 Java IFS source file . . . . . . > '/javacmds/IBMiSamples/HelloWorld.java'



 Java class path  . . . . . . . . > '.'



                                                                  ...
 Java virtual machine to use  . .   6032          4064, 5032, 5064, 6032, 6064
 Display standard output result   > *YES          *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
 Prt java std output on errors  .   *NO           *NO, *YES

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```



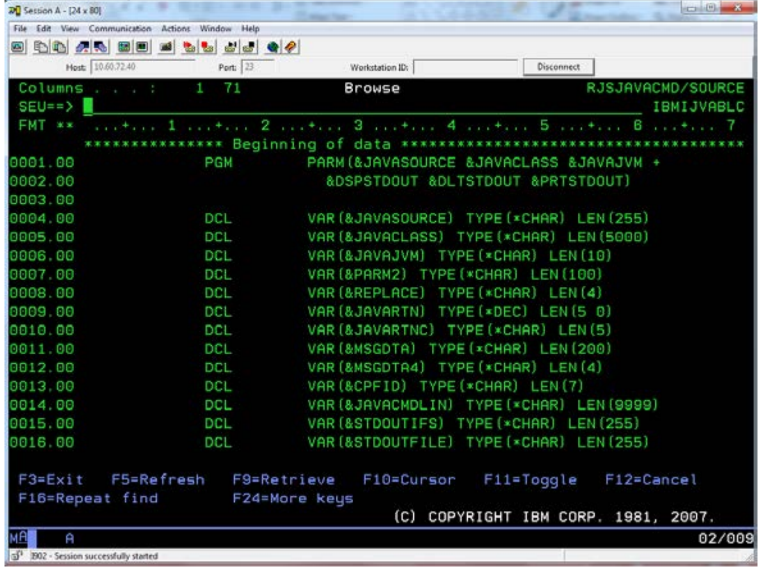
```
 Columns . . . :    1  71           Browse                  RJSJAVACMD/SOURCE
 SEU==>                                                            IBMIJVABLC
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data *************************************
                       PGM        PARM(&JAVASOURCE &JAVACLASS &JAVAJVM +
                                    &DSPSTDOUT &DLTSTDOUT &PRTSTDOUT)

                       DCL        VAR(&JAVASOURCE) TYPE(*CHAR) LEN(255)
                       DCL        VAR(&JAVACLASS) TYPE(*CHAR) LEN(5000)
                       DCL        VAR(&JAVAJVM) TYPE(*CHAR) LEN(10)
                       DCL        VAR(&PARM2) TYPE(*CHAR) LEN(100)
                       DCL        VAR(&REPLACE) TYPE(*CHAR) LEN(4)
                       DCL        VAR(&JAVARTN) TYPE(*DEC) LEN(5 0)
                       DCL        VAR(&JAVARTNC) TYPE(*CHAR) LEN(5)
                       DCL        VAR(&MSGDTA) TYPE(*CHAR) LEN(200)
                       DCL        VAR(&MSGDTA4) TYPE(*CHAR) LEN(4)
                       DCL        VAR(&CPFID) TYPE(*CHAR) LEN(7)
                       DCL        VAR(&JAVACMDLIN) TYPE(*CHAR) LEN(9999)
                       DCL        VAR(&STDOUTIFS) TYPE(*CHAR) LEN(255)
                       DCL        VAR(&STDOUTFILE) TYPE(*CHAR) LEN(255)

 F3=Exit   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle   F12=Cancel
 F16=Repeat find        F24=More keys
                                        (C) COPYRIGHT IBM CORP. 1981, 2007.
```


Type CL Command or Embed in a CL Program
CL Command
Command Processing Program CL, RPG, COBOL
Parameters or Data Returned to Calling CL Program or to Command Line as CPF Messages

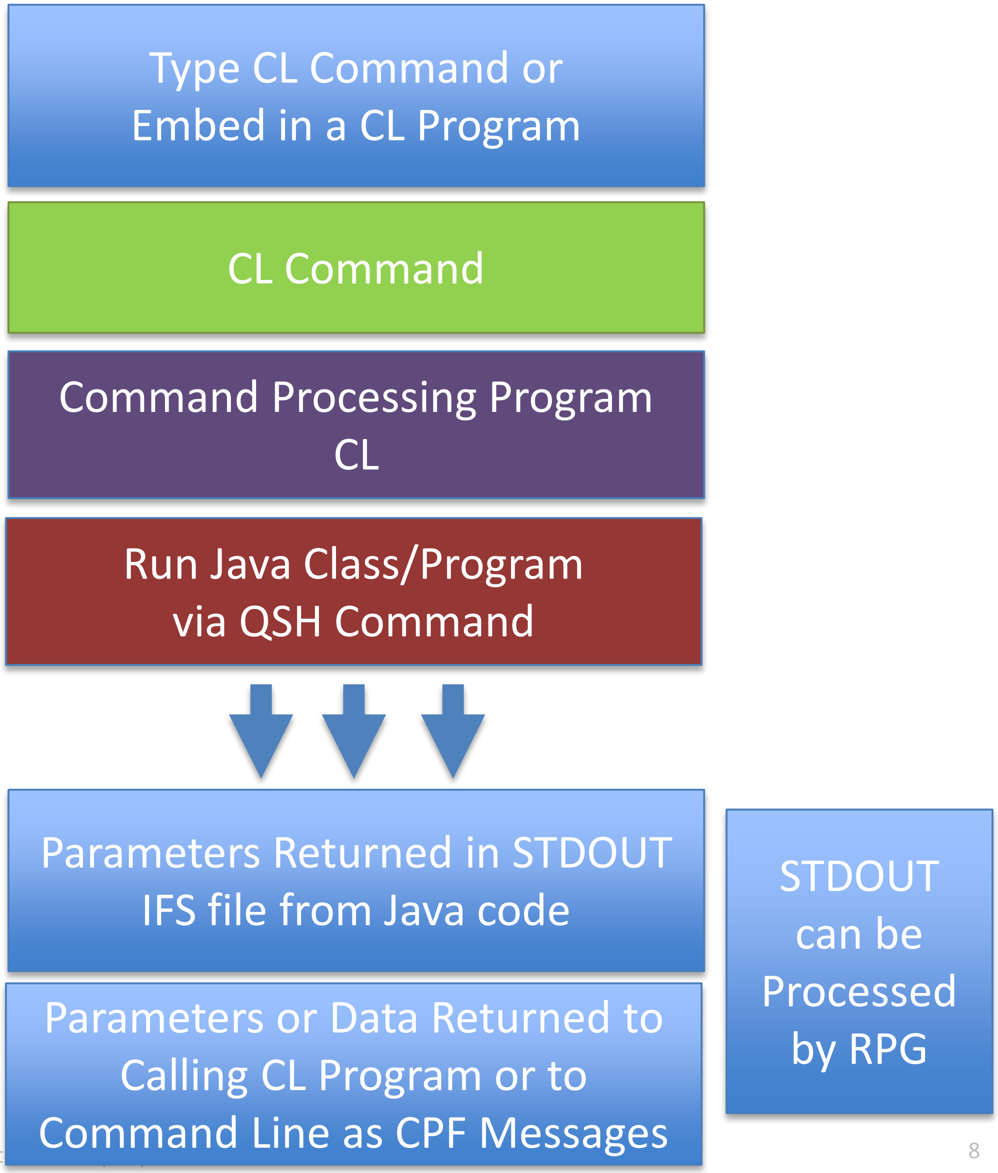
CL Command Process Structure using Java



```
                     Compile Java Program Source (IBMIJVABLD)

 Type choices, press Enter.

 Java IFS source file . . . . . . > '/javacmds/IBMiSamples/HelloWorld.java'



 Java class path  . . . . . . . . > '.'






                                                                          ...
 Java virtual machine to use  . .   6032          4064, 5032, 5064, 6032, 6064
 Display standard output result   > *YES          *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
 Prt java std output on errors  .   *NO           *NO, *YES

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```



```
 Columns . . . :    1  71            Browse                  RJSJAVACMD/SOURCE
 SEU==>                                                              IBMIJVABLC
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data *************************************
0001.00              PGM        PARM(&JAVASOURCE &JAVACLASS &JAVAJVM +
0002.00                           &DSPSTDOUT &DLTSTDOUT &PRTSTDOUT)
0003.00
0004.00              DCL        VAR(&JAVASOURCE) TYPE(*CHAR) LEN(255)
0005.00              DCL        VAR(&JAVACLASS) TYPE(*CHAR) LEN(5000)
0006.00              DCL        VAR(&JAVAJVM) TYPE(*CHAR) LEN(10)
0007.00              DCL        VAR(&PARM2) TYPE(*CHAR) LEN(100)
0008.00              DCL        VAR(&REPLACE) TYPE(*CHAR) LEN(4)
0009.00              DCL        VAR(&JAVARTN) TYPE(*DEC) LEN(5 0)
0010.00              DCL        VAR(&JAVARTNC) TYPE(*CHAR) LEN(5)
0011.00              DCL        VAR(&MSGDTA) TYPE(*CHAR) LEN(200)
0012.00              DCL        VAR(&MSGDTA4) TYPE(*CHAR) LEN(4)
0013.00              DCL        VAR(&CPFID) TYPE(*CHAR) LEN(7)
0014.00              DCL        VAR(&JAVACMDLIN) TYPE(*CHAR) LEN(9999)
0015.00              DCL        VAR(&STDOUTIFS) TYPE(*CHAR) LEN(255)
0016.00              DCL        VAR(&STDOUTFILE) TYPE(*CHAR) LEN(255)

 F3=Exit   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle   F12=Cancel
 F16=Repeat find        F24=More keys
                                          (C) COPYRIGHT IBM CORP. 1981, 2007.
```



Type CL Command or Embed in a CL Program
CL Command
Command Processing Program CL
Run Java Class/Program via QSH Command
Parameters Returned in STDOUT IFS file from Java code
Parameters or Data Returned to Calling CL Program or to Command Line as CPF Messages
STDOUT can be Processed by RPG

Benefits of Java Approach

- JVM startup time is very fast (especially 32-bit)
- Wide range of open source Java APIs available
- Don't have to deal with oddities of embedding Java in RPG

  Mapping methods, single instance of JVM per job, classpath oddities and debugging.
- Can switch JVM between Java calls if needed
- Can do some useful/re-usable IBM i CL command development
- Adds to your marketable skillset

Potential Drawbacks of Java Approach

- Qshell runs a new JVM instance for each call
- Java is invisibly spawned in a new process
- May want to write NEP style Java app to read data queue or table if performance is an issue
- You DO have to learn some Java
- Need to benchmark if you plan to do thousands of calls per day
- You may want to compile CL as CLP, not CLLE

  ** I've had odd unexplainable data corruptions over the years that were fixed by simply changing source type from CLLE to CLP

Installing RJSJAVACMD Library & IFS Objects

- Download library from RJS and run exe to unzip and install library

  http://downloads.rjssoftware.com/files/classes/2015/rjsjavacmd.exe

- Build objects after restore

```
// Run the following commands to build CL/RPG programs
ADDLIBLE RJSJAVACMD
CRTCLPGM PGM(RJSJAVACMD/IBMIBUILDC) SRCFILE(RJSJAVACMD/SOURCE) SRCMBR(IBMIBUILDC) REPLACE(*YES)
// Build CL and RPG
CALL IBMIBUILDC
// Call program to restore IFS objects /javacmds/ibmisamples
CALL RSTIFS1
// Call java build CL program to build java source
CALL JAVABUILD1
```

Slides

http://downloads.rjssoftware.com/files/classes/2015/rjsjavacmd.pdf

Agenda

- Why use Java and CL Together
- **Review development environments for Java.**
- **Code, compile and build**
- Work through sample commands

Development Environments for Java

- Rational Developer for i or any Eclipse editor - Best
- Green screen editing via WRKLNK and EDTF / Qshell compile
- Green screen via SEU / Qshell compile
- Notepad ++ or other PC Editor and Copy to IFS and compile
- Other odd combinations perhaps ?

Developing Java Programs from Green Screen

helpsystems

- Familiar with 5250 green screen
- Edit via EDTF or SEU
- Compile and test in IFS
- Deploy .class files to IFS
- Practical ?

Editing Java with 5250 WRKLNK and EDTF

- Work with IFS Source **WRKLNK OBJ('/javacmds/ibmisamples/*')**

- Use **Option 2** to edit the **hello1.java** file via the EDTF command.

- Compile java program

```
STRQSH
cd /javacmds/ibmisamples
javac hello1.java
```

- Run java program (Don't specify hello1.class)

```
java hello
```

- Java standard output

```
START: Program Start
Test1
Test2
OK: This was a successful program run
END: Program End
$
```

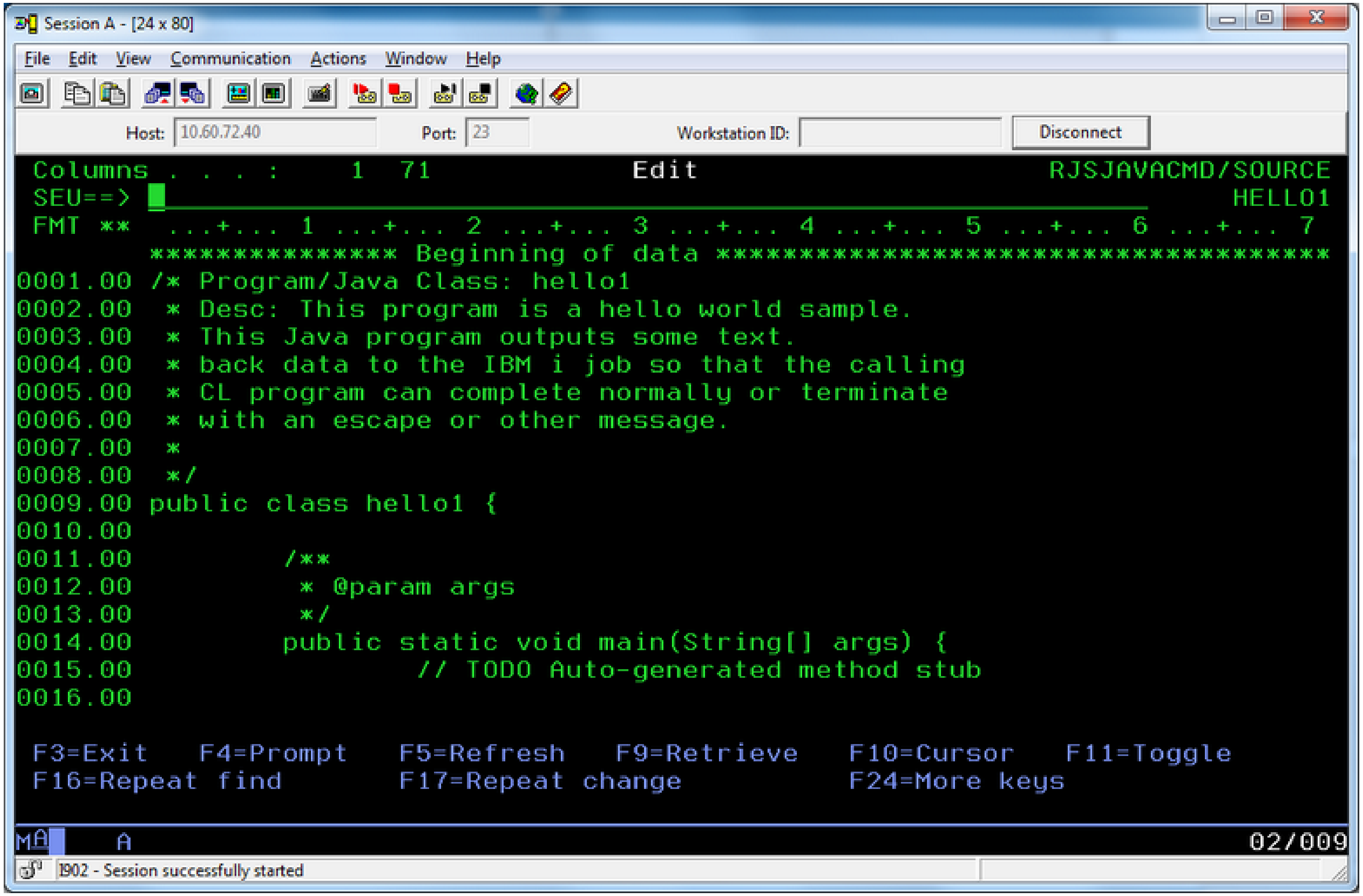
Editing Java with 5250 SEU

- Copy source from IFS to source file in library.
- Edit source member with SEU.
- Copy source from source file back to IFS
- Build and compile the java from IFS
- Not pretty but it works for those who won’t use RDI or Eclipse

Copy Java Source File from IFS to Library

```
CPYFRMSTMF FROMSTMF('/javacmds/ibmisamples/hello1.java')
        TOMBR('/qsys.lib/rjsjavacmd.lib/source.file/hello1.mbr')
        MBROPT(*REPLACE)
        CVTDTA(*AUTO)
        STMFCCSID(*STMF)
        DBFCCSID(*FILE)
        ENDLINFMT(*ALL)
        TABEXPN(*YES)
        STMFCODPAG(*STMF)
```

Edit and Save Java Source with SEU



```
 Columns . . . :     1  71              Edit                    RJSJAVACMD/SOURCE
 SEU==>                                                                    HELLO1
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data *************************************
0001.00 /* Program/Java Class: hello1
0002.00  * Desc: This program is a hello world sample.
0003.00  * This Java program outputs some text.
0004.00  * back data to the IBM i job so that the calling
0005.00  * CL program can complete normally or terminate
0006.00  * with an escape or other message.
0007.00  *
0008.00  */
0009.00 public class hello1 {
0010.00
0011.00         /**
0012.00          * @param args
0013.00          */
0014.00         public static void main(String[] args) {
0015.00                 // TODO Auto-generated method stub
0016.00

 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle
 F16=Repeat find       F17=Repeat change          F24=More keys

MA      A                                                                 02/009
```

I902 - Session successfully started

Copy Java Source File from Library to IFS

```
CPYTOSTMF FROMMBR('/qsys.lib/rjsjavacmd.lib/source.file/hello1.mbr')
          TOSTMF('/javacmds/ibmisamples/hello1.java')
          STMFOPT(*REPLACE)
          CVTDTA(*AUTO)
          DBFCCSID(*FILE)
          STMFCCSID(437)
          ENDLINFMT(*CRLF)
          AUT(*DFT)
```

Compiling Java Source from IFS

- Compile java program

```
STRQSH
cd /javacmds/ibmisamples
javac hello1.java
```

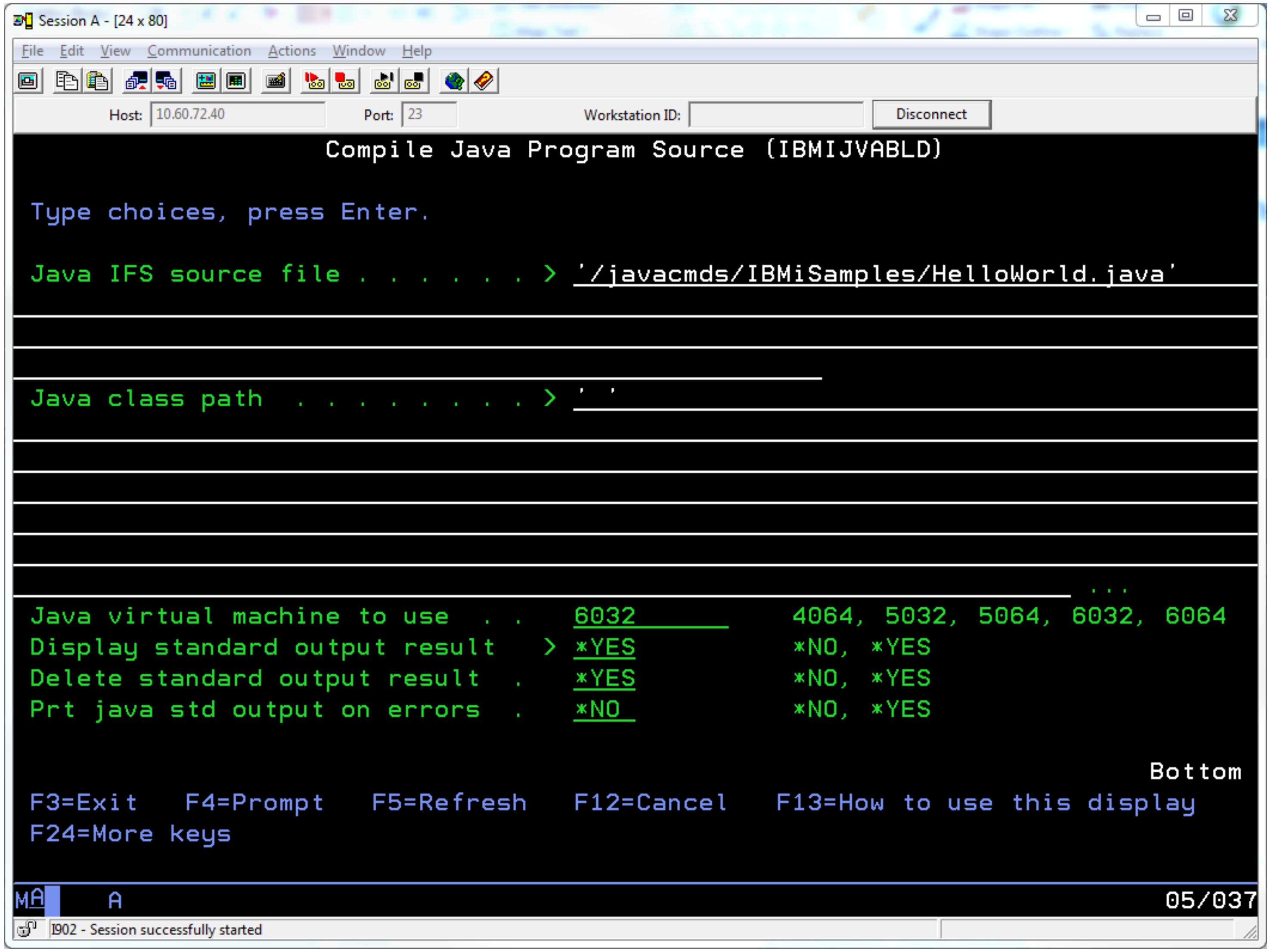
- Run java program (Don’t specify hello1.class)

```
java hello1
```

- Java standard output

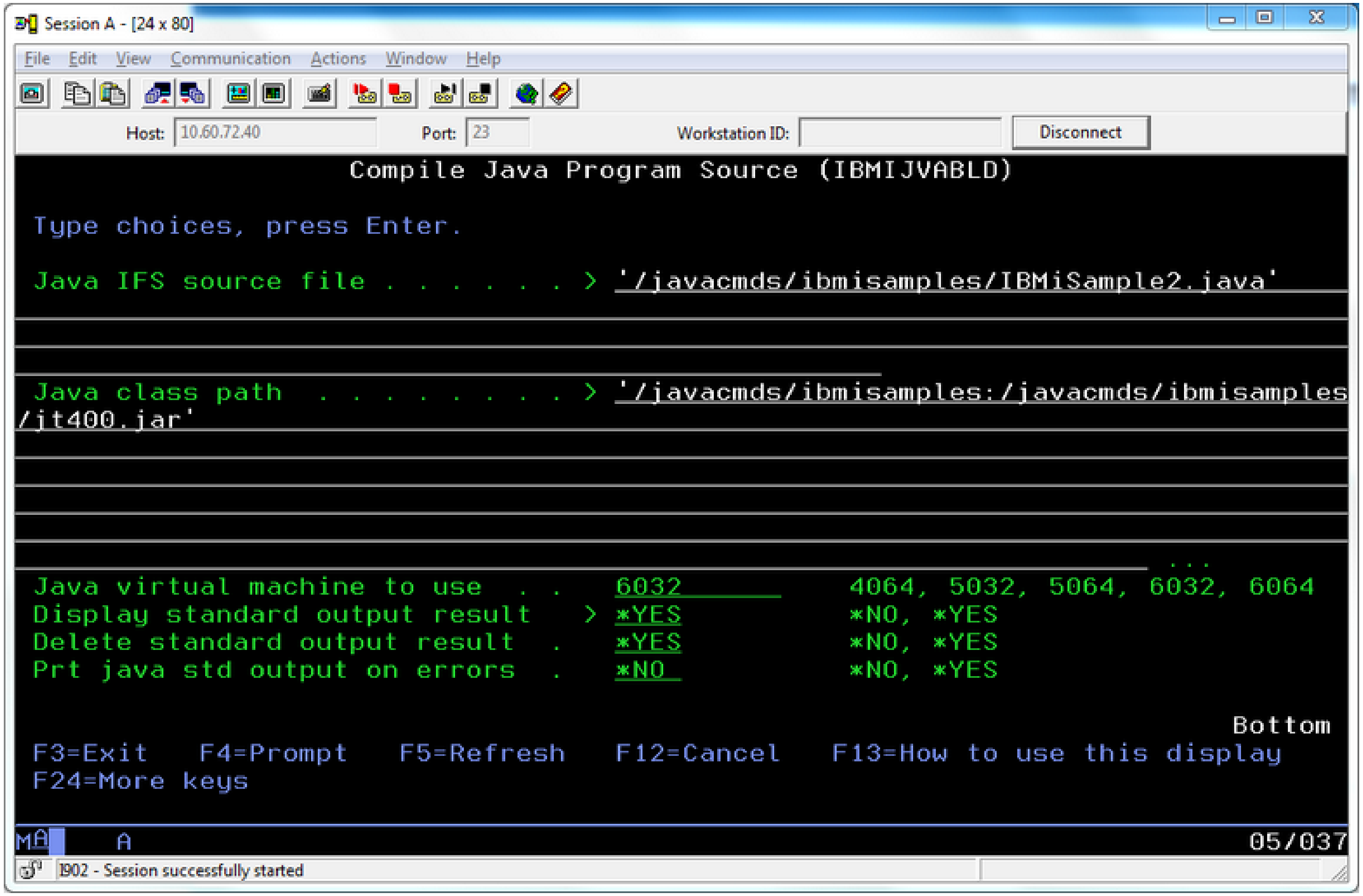
```
START: Program Start
Test1
Test2
OK: This was a successful program run
END: Program End
$
```

Compiling Java from Green Screen Helper



```
                    Compile Java Program Source (IBMIJVABLD)

 Type choices, press Enter.

 Java IFS source file . . . . . . > '/javacmds/IBMiSamples/HelloWorld.java'




 Java class path  . . . . . . . . > ' '








 Java virtual machine to use  . .   6032          4064, 5032, 5064, 6032, 6064
 Display standard output result   > *YES          *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
 Prt java std output on errors  .   *NO           *NO, *YES

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     A                                                                05/037
```

Compiling Java with Jar API References



```
                  Compile Java Program Source (IBMIJVABLD)

 Type choices, press Enter.

 Java IFS source file . . . . . . > '/javacmds/ibmisamples/IBMiSample2.java'




 Java class path  . . . . . . . . > '/javacmds/ibmisamples:/javacmds/ibmisamples
/jt400.jar'





                                                                          ...
 Java virtual machine to use  . .   6032          4064, 5032, 5064, 6032, 6064
 Display standard output result   > *YES          *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
 Prt java std output on errors  .   *NO           *NO, *YES

                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA     A                                                                05/037
```

I902 - Session successfully started

Compiling Java from Green Screen Helper

- Compile java programs to /javacmds/IBMiSamples
- Select JVM to use **6032** means Java 1.6 32-Bit
- Use DSPSTDOUT to view any errors.
- Use PRTSTDOUT to view any errors.
- Use DLTSTDOUT to delete STDOUT IFS file from /rjstemp
- IBMIJVABLD JAVASOURCE('/javacmds/IBMiSamples/hello1.java') CLASSPATH(' ') JVM(6032) DSPSTDOUT(*YES) DLTSTDOUT(*YES) PRTSTDOUT(*NO)

Running via RUNJVA Command or Qshell

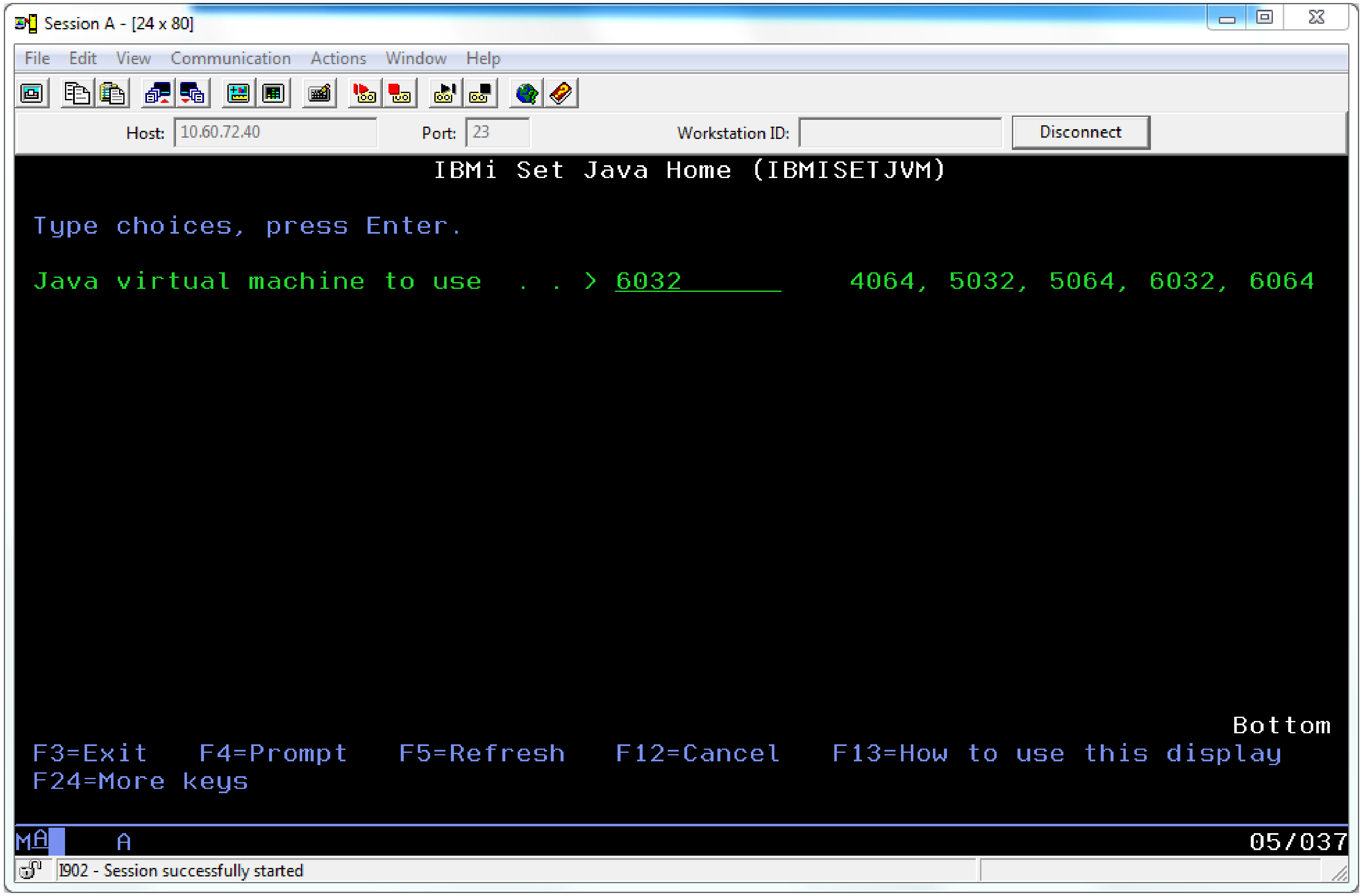
- Run from regular command line
- RUNJVA CLASS('hello1') CLASSPATH('/javacmds/ibmisamples')
- Run from QSHELL.

  STRQSH

  cd /javacmds/ibmisamples

  java hello1
- Qshell compile and run preferred - More control
- Our CL examples will all use Qshell

IBMISETJVM - Set JAVA_HOME to Select JVM

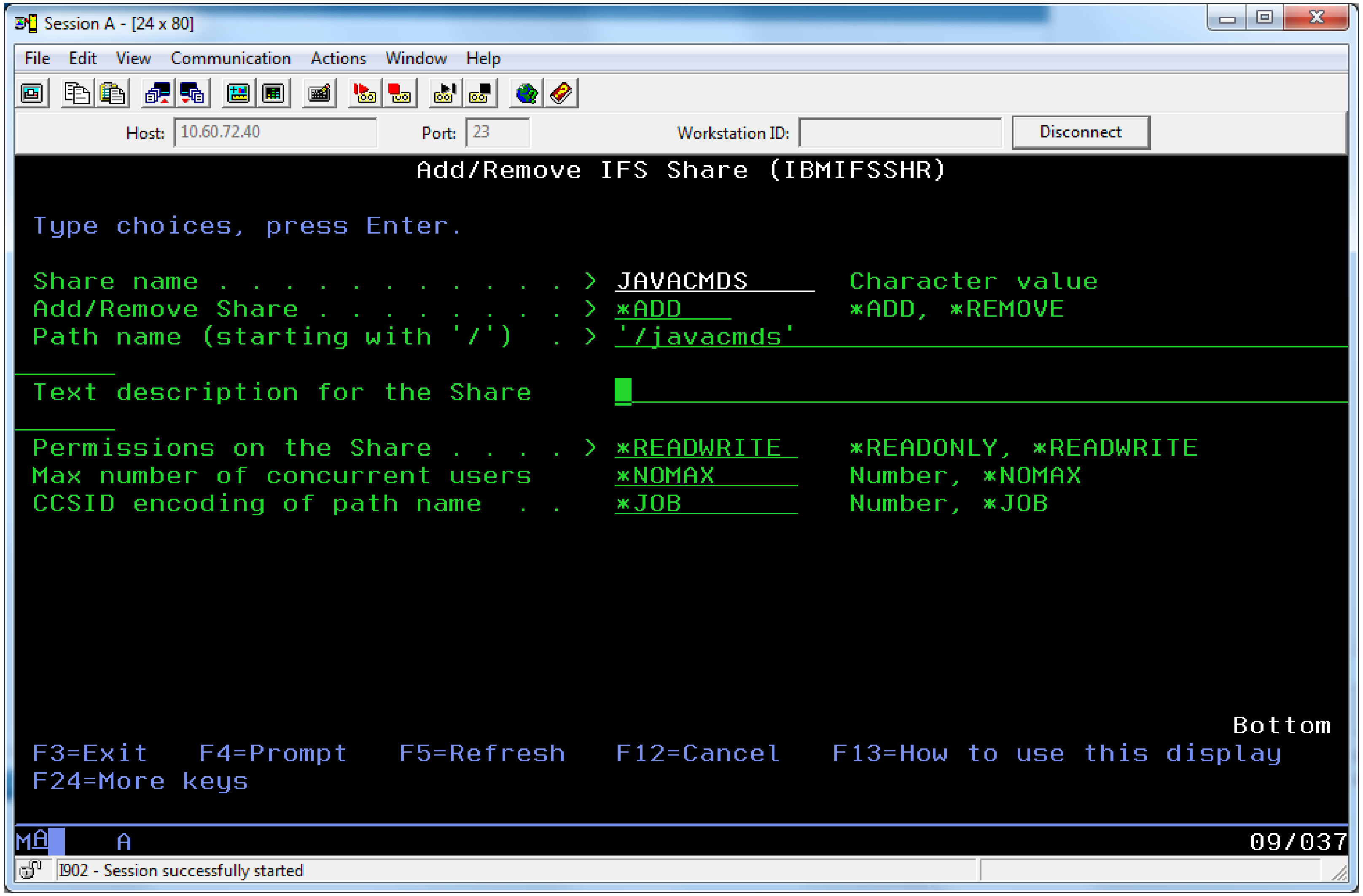


Host: 10.60.72.40 Port: 23 Workstation ID:

```
                        IBMi Set Java Home (IBMISETJVM)

 Type choices, press Enter.

 Java virtual machine to use . . > 6032          4064, 5032, 5064, 6032, 6064

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     A                                                               05/037
```

I902 - Session successfully started

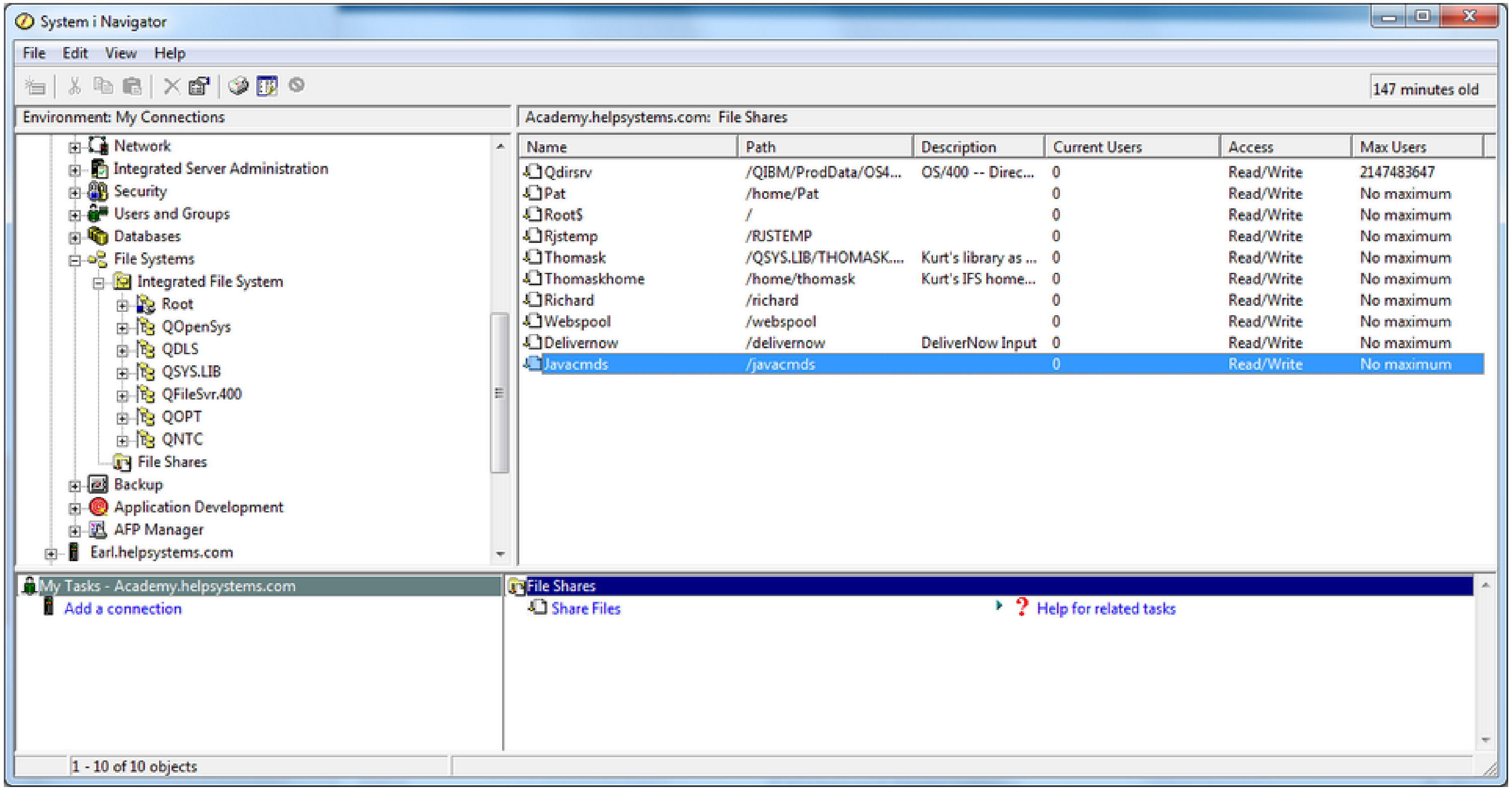
IBMIFSSHR - Set Up NetServer Dir Share



```
                        Add/Remove IFS Share (IBMIFSSHR)

 Type choices, press Enter.

 Share name . . . . . . . . . . . > JAVACMDS      Character value
 Add/Remove Share . . . . . . . . > *ADD          *ADD, *REMOVE
 Path name (starting with '/')  . > '/javacmds'

 Text description for the Share

 Permissions on the Share . . . . > *READWRITE    *READONLY, *READWRITE
 Max number of concurrent users     *NOMAX        Number, *NOMAX
 CCSID encoding of path name  . .   *JOB          Number, *JOB






                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys

MA     A                                                                09/037
```

I902 - Session successfully started

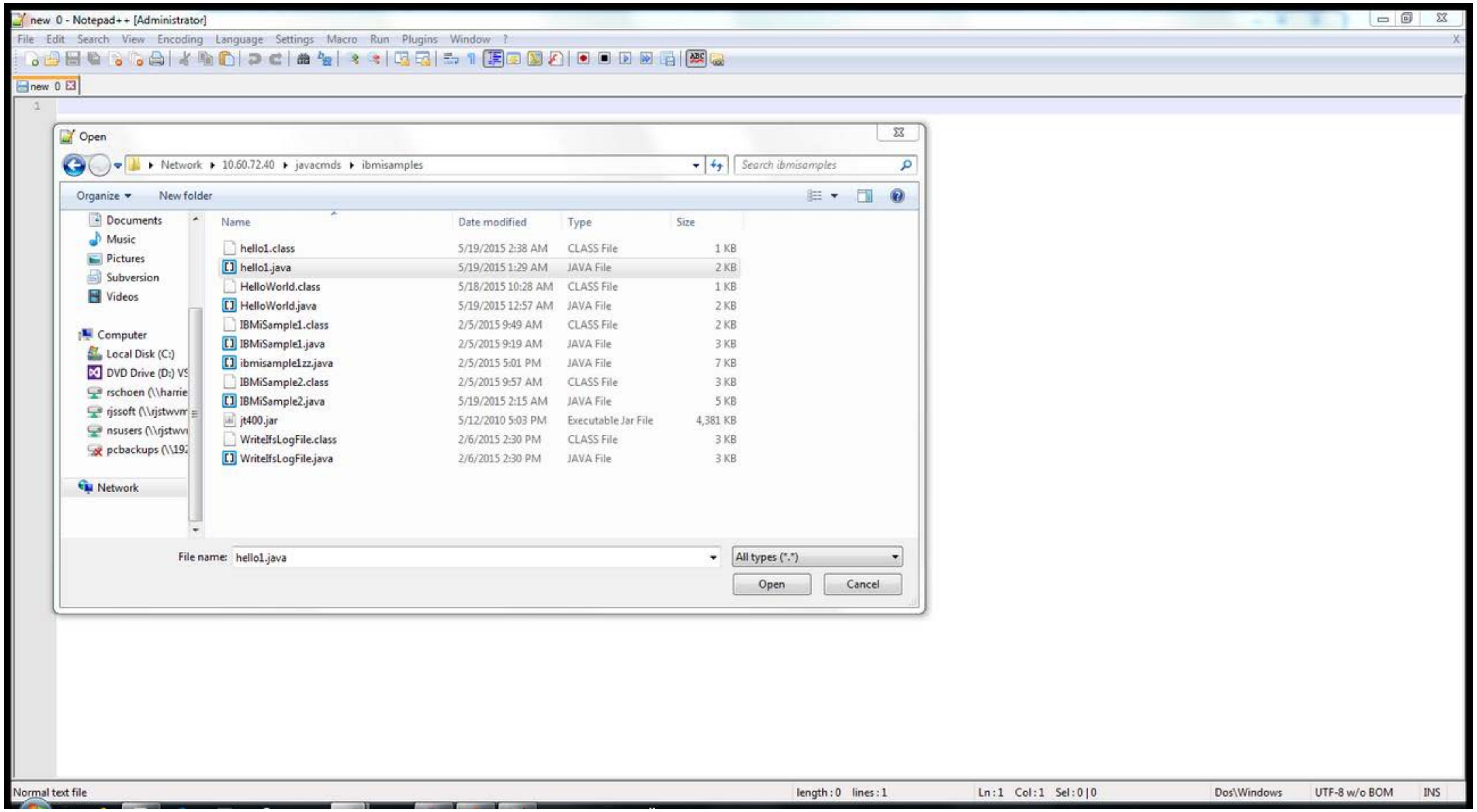
Viewing Available Shares in IBM i Navigator



Academy.helpsystems.com: File Shares

Name	Path	Description	Current Users	Access	Max Users
Qdirsrv	/QIBM/ProdData/OS4...	OS/400 -- Direc...	0	Read/Write	2147483647
Pat	/home/Pat		0	Read/Write	No maximum
Root$	/		0	Read/Write	No maximum
Rjstemp	/RJSTEMP		0	Read/Write	No maximum
Thomask	/QSYS.LIB/THOMASK....	Kurt's library as ...	0	Read/Write	No maximum
Thomaskhome	/home/thomask	Kurt's IFS home...	0	Read/Write	No maximum
Richard	/richard		0	Read/Write	No maximum
Webspool	/webspool		0	Read/Write	No maximum
Delivernow	/delivernow	DeliverNow Input	0	Read/Write	No maximum
Javacmds	/javacmds		0	Read/Write	No maximum

Editing from IFS Share with Notepad++



new 0 - Notepad++ [Administrator]

Network ▸ 10.60.72.40 ▸ javacmds ▸ ibmisamples

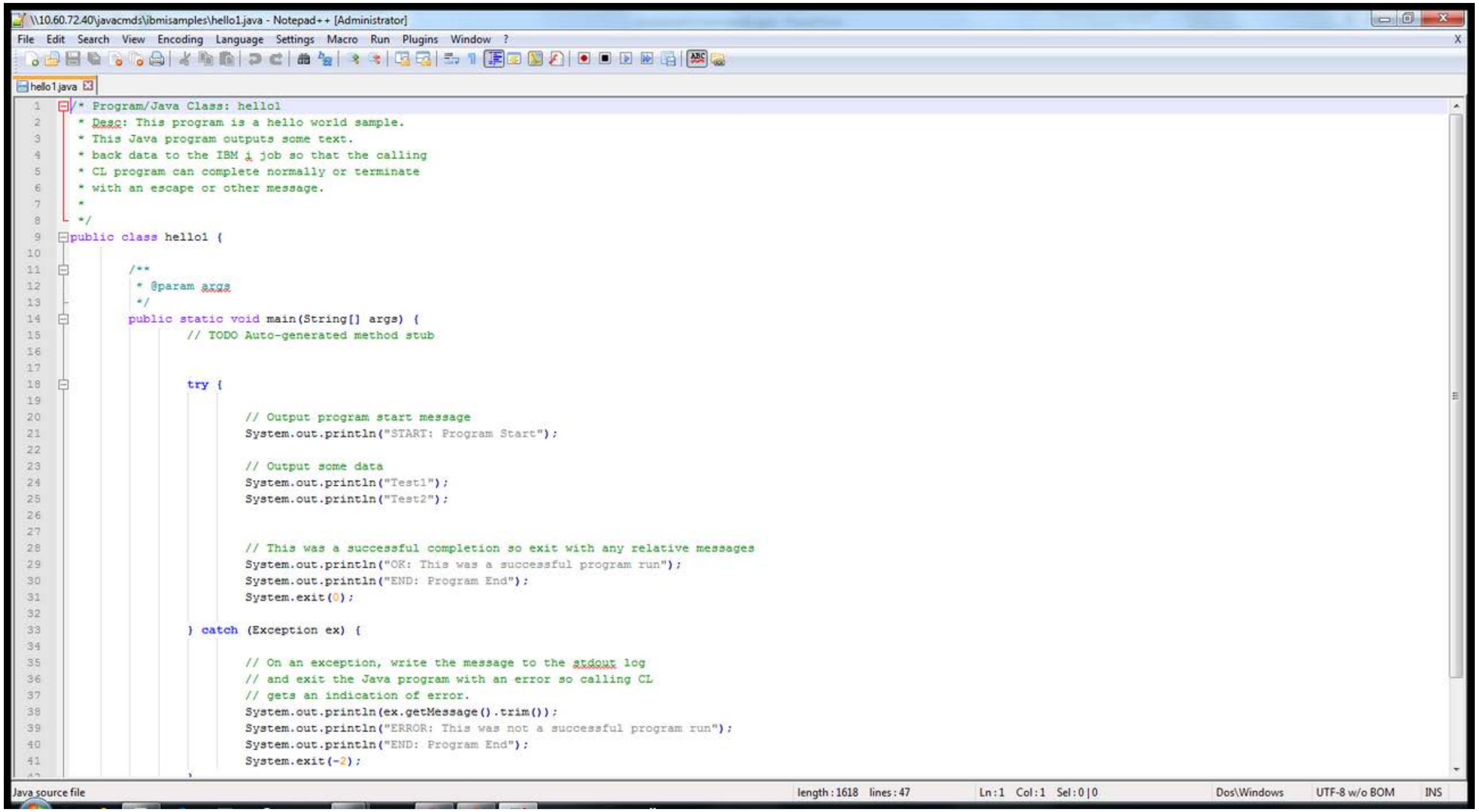
Name	Date modified	Type	Size
hello1.class	5/19/2015 2:38 AM	CLASS File	1 KB
hello1.java	5/19/2015 1:29 AM	JAVA File	2 KB
HelloWorld.class	5/18/2015 10:28 AM	CLASS File	1 KB
HelloWorld.java	5/19/2015 12:57 AM	JAVA File	2 KB
IBMiSample1.class	2/5/2015 9:49 AM	CLASS File	2 KB
IBMiSample1.java	2/5/2015 9:19 AM	JAVA File	3 KB
ibmisample1zz.java	2/5/2015 5:01 PM	JAVA File	7 KB
IBMiSample2.class	2/5/2015 9:57 AM	CLASS File	3 KB
IBMiSample2.java	5/19/2015 2:15 AM	JAVA File	5 KB
jt400.jar	5/12/2010 5:03 PM	Executable Jar File	4,381 KB
WriteIfsLogFile.class	2/6/2015 2:30 PM	CLASS File	3 KB
WriteIfsLogFile.java	2/6/2015 2:30 PM	JAVA File	3 KB

File name: hello1.java

All types (*.*)

Normal text file length : 0 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

Editing from IFS Share with Notepad++

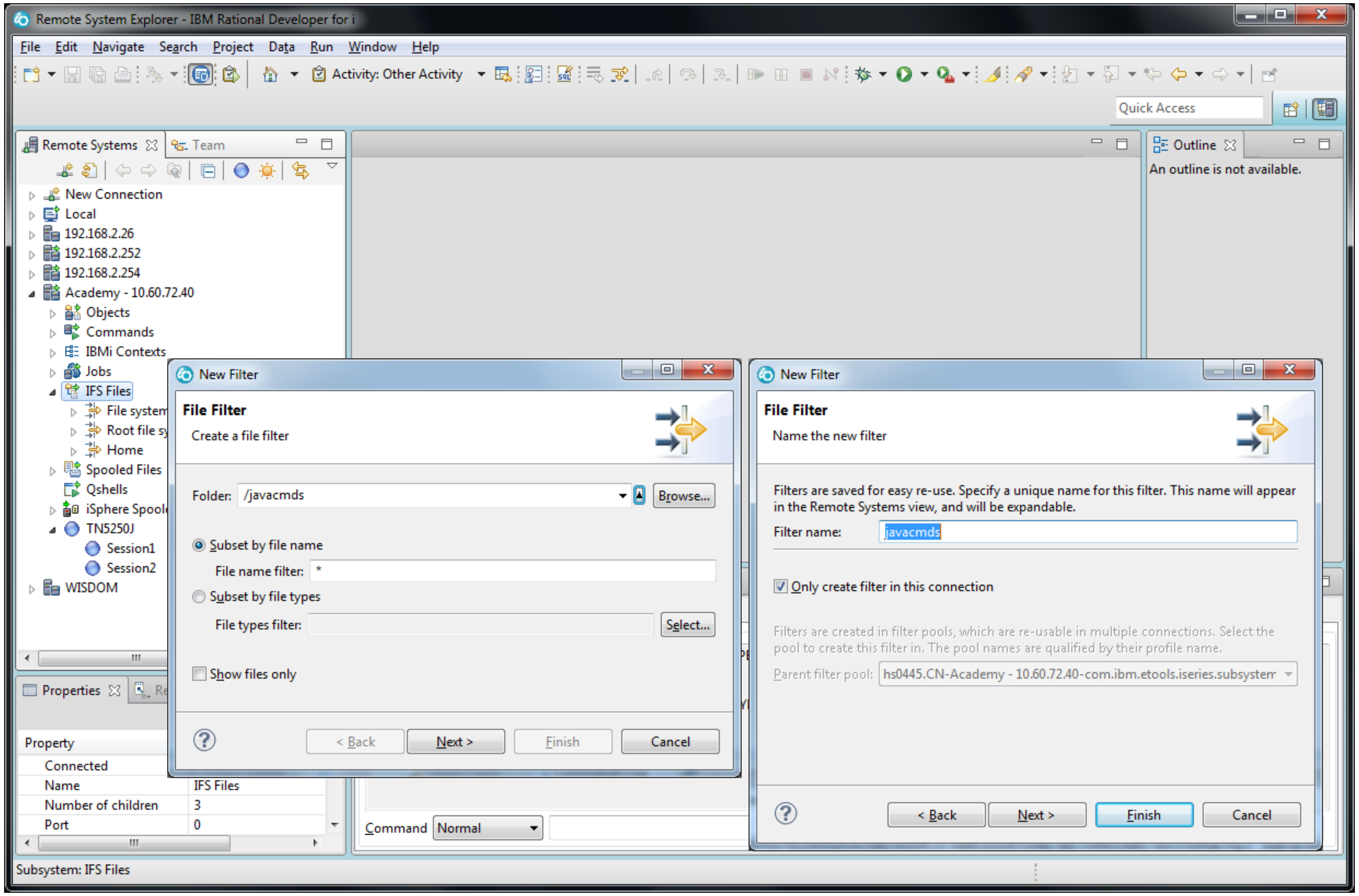

\\10.60.72.40\javacmds\ibmisamples\hello1.java - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
hello1.java

```
/* Program/Java Class: hello1
 * Desc: This program is a hello world sample.
 * This Java program outputs some text.
 * back data to the IBM i job so that the calling
 * CL program can complete normally or terminate
 * with an escape or other message.
 *
 */
public class hello1 {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub


		try {

			// Output program start message
			System.out.println("START: Program Start");

			// Output some data
			System.out.println("Test1");
			System.out.println("Test2");


			// This was a successful completion so exit with any relative messages
			System.out.println("OK: This was a successful program run");
			System.out.println("END: Program End");
			System.exit(0);

		} catch (Exception ex) {

			// On an exception, write the message to the stdout log
			// and exit the Java program with an error so calling CL
			// gets an indication of error.
			System.out.println(ex.getMessage().trim());
			System.out.println("ERROR: This was not a successful program run");
			System.out.println("END: Program End");
			System.exit(-2);
```

Developing Java Programs in RDI

- Graphical editing. Eclipse is FREE if you don’t have RDI
- Compile and debug locally on PC, even without IBM i
- Deploy .class files to IFS via copy/paste and RSE or file share

Rational Developer RSE IFS Filter on /javacmds



Remote System Explorer - IBM Rational Developer for i

Remote Systems
- New Connection
- Local
- 192.168.2.26
- 192.168.2.252
- 192.168.2.254
- Academy - 10.60.72.40
  - Objects
  - Commands
  - IBMi Contexts
  - Jobs
  - IFS Files
    - File system
    - Root file sy
    - Home
  - Spooled Files
  - Qshells
  - iSphere Spool
  - TN5250J
    - Session1
    - Session2
- WISDOM

Property	
Connected	
Name	IFS Files
Number of children	3
Port	0

Subsystem: IFS Files

New Filter

File Filter

Create a file filter

Folder: /javacmds

(●) Subset by file name

File name filter: *

( ) Subset by file types

File types filter:

[ ] Show files only

New Filter

File Filter

Name the new filter

Filters are saved for easy re-use. Specify a unique name for this filter. This name will appear in the Remote Systems view, and will be expandable.

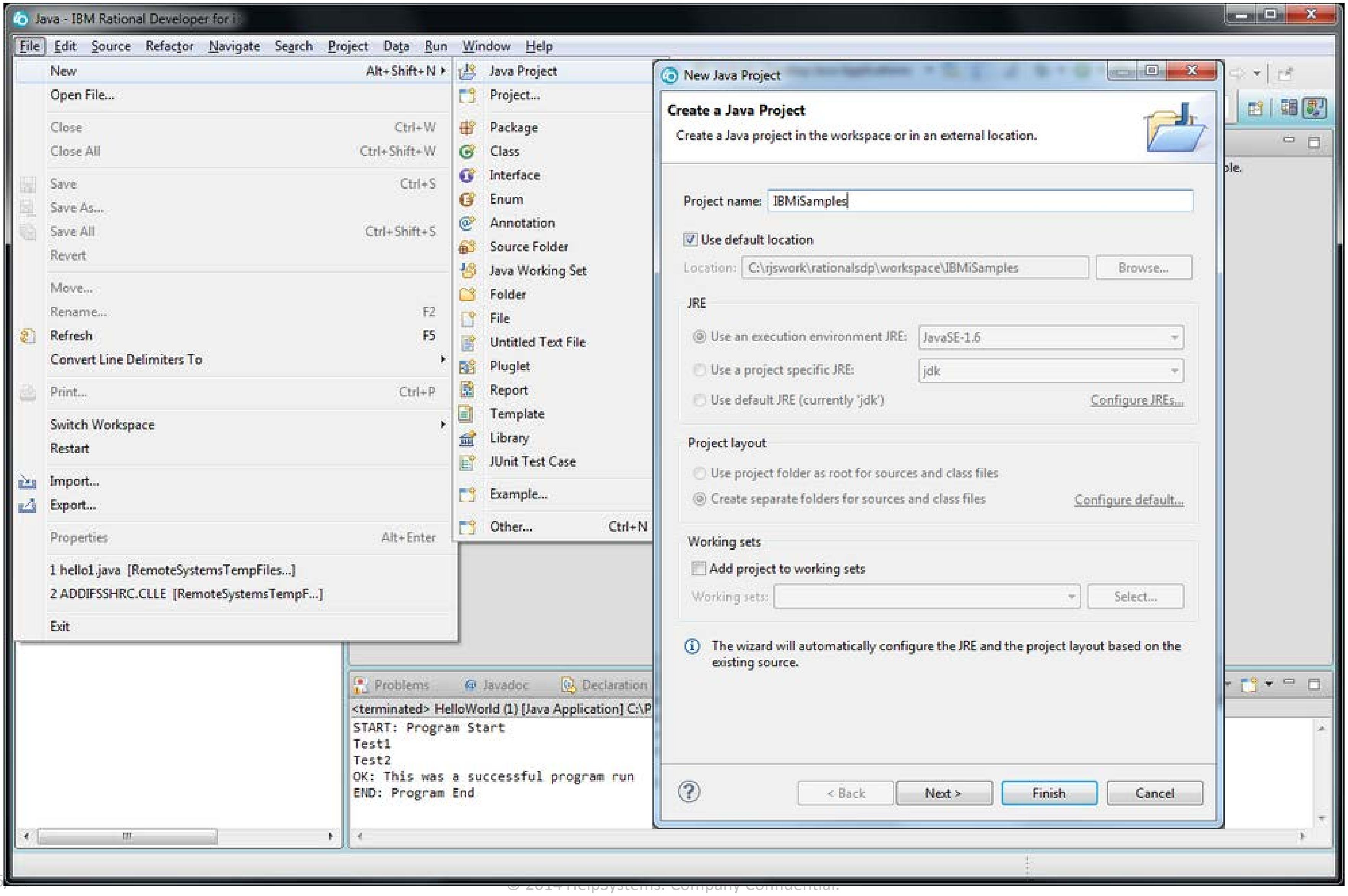
Filter name: javacmds

[✓] Only create filter in this connection

Filters are created in filter pools, which are re-usable in multiple connections. Select the pool to create this filter in. The pool names are qualified by their profile name.

Parent filter pool: hs0445.CN-Academy - 10.60.72.40-com.ibm.etools.iseries.subsystem

Start A New Java Project in RDI



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: IBMiSamples

[✓] Use default location

Location: C:\rjswork\rationalsdp\workspace\IBMiSamples   Browse...

JRE

(●) Use an execution environment JRE: JavaSE-1.6

( ) Use a project specific JRE: jdk

( ) Use default JRE (currently 'jdk')   Configure JREs...

Project layout

( ) Use project folder as root for sources and class files

(●) Create separate folders for sources and class files   Configure default...

Working sets

[ ] Add project to working sets

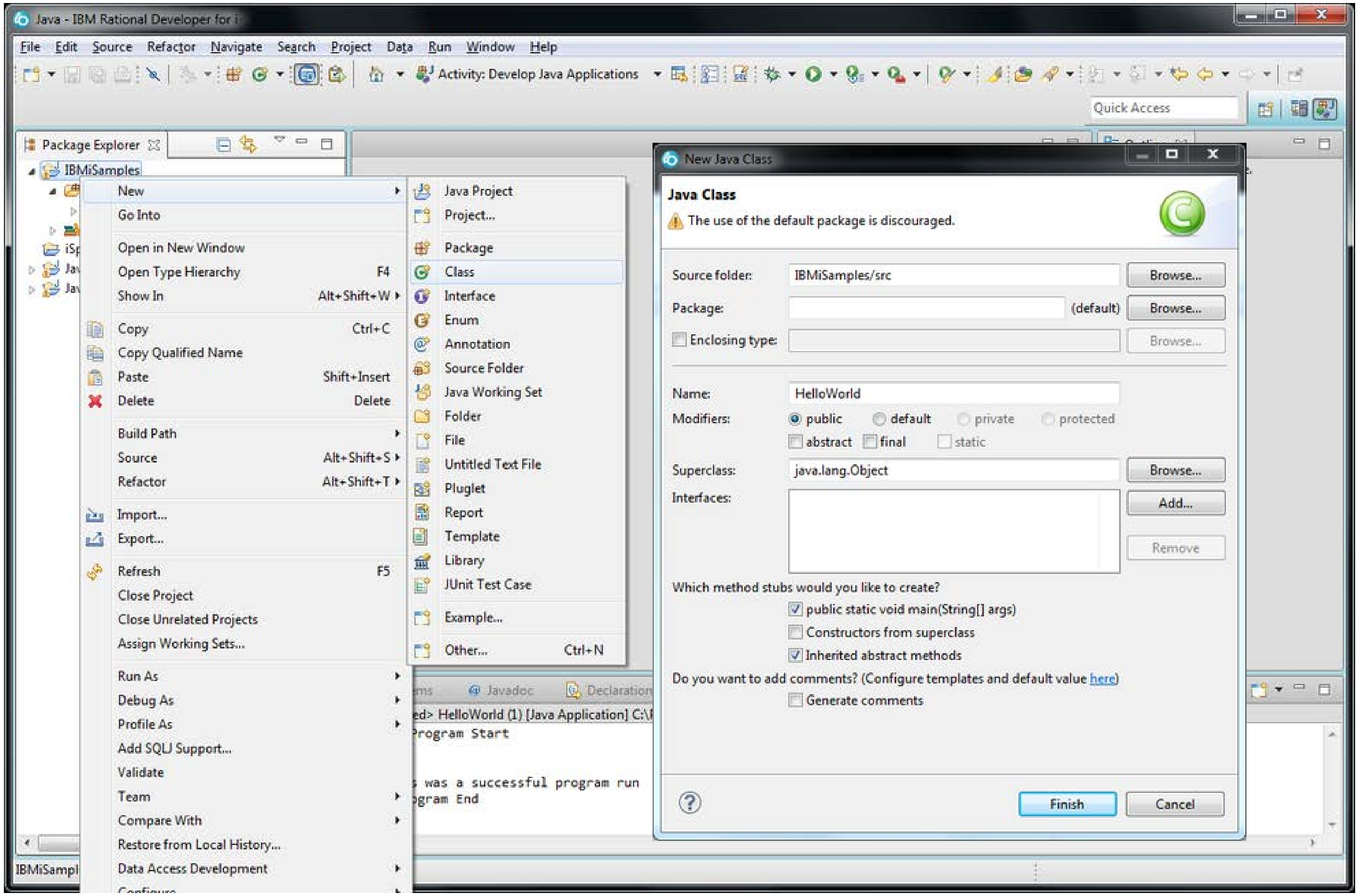
Working sets: Select...

The wizard will automatically configure the JRE and the project layout based on the existing source.

< Back   Next >   Finish   Cancel

```
<terminated> HelloWorld (1) [Java Application] C:\P
START: Program Start
Test1
Test2
OK: This was a successful program run
END: Program End
```

Create the Hello World Class



The use of the default package is discouraged.

Source folder: IBMiSamples/src

Package: (default)

Enclosing type: ☐

Name: HelloWorld

Modifiers: [✓ public] default private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

☐ Constructors from superclass

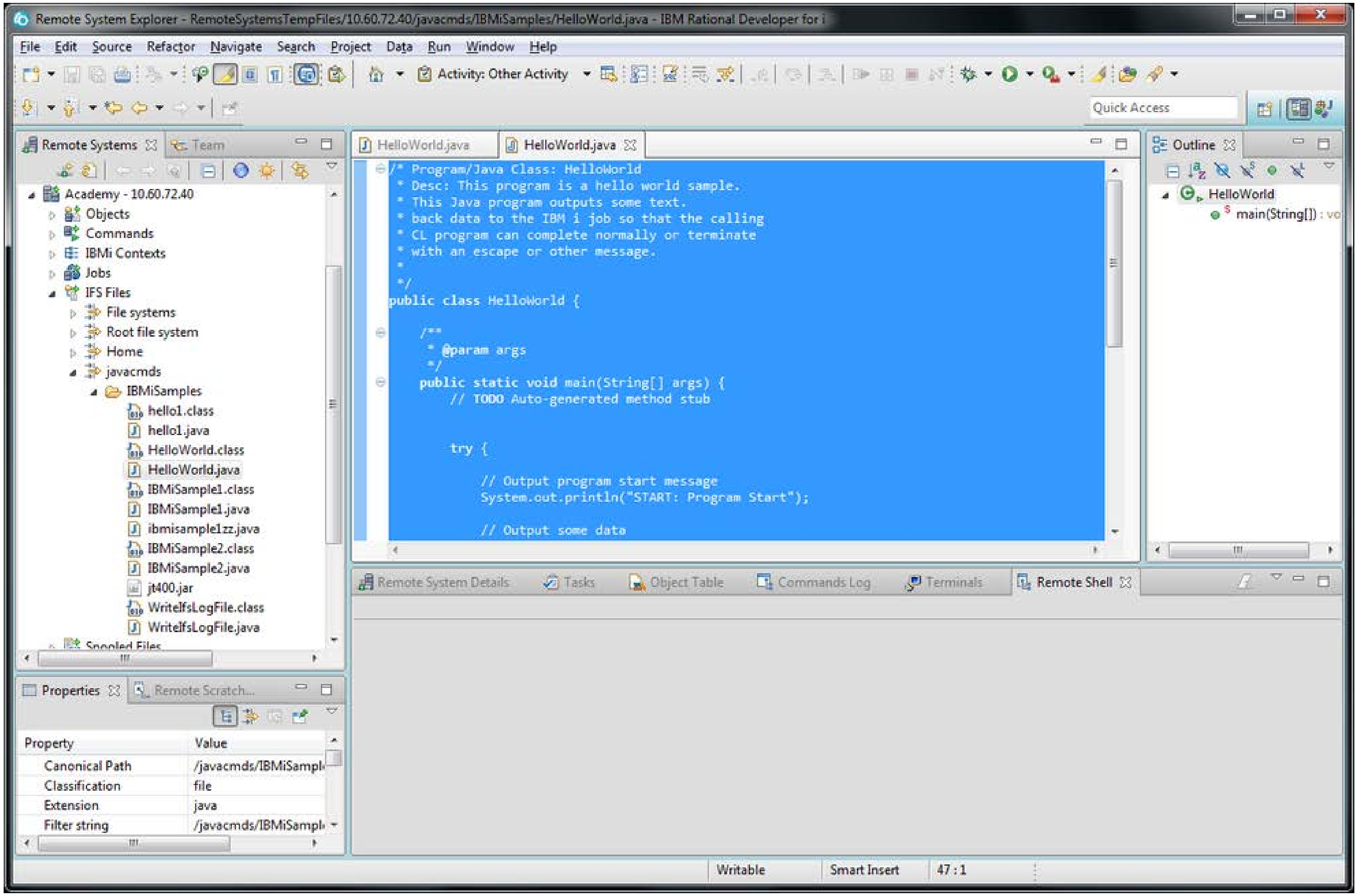
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

```
ed> HelloWorld (1) [Java Application] C:\
rogram Start

 was a successful program run
ogram End
```

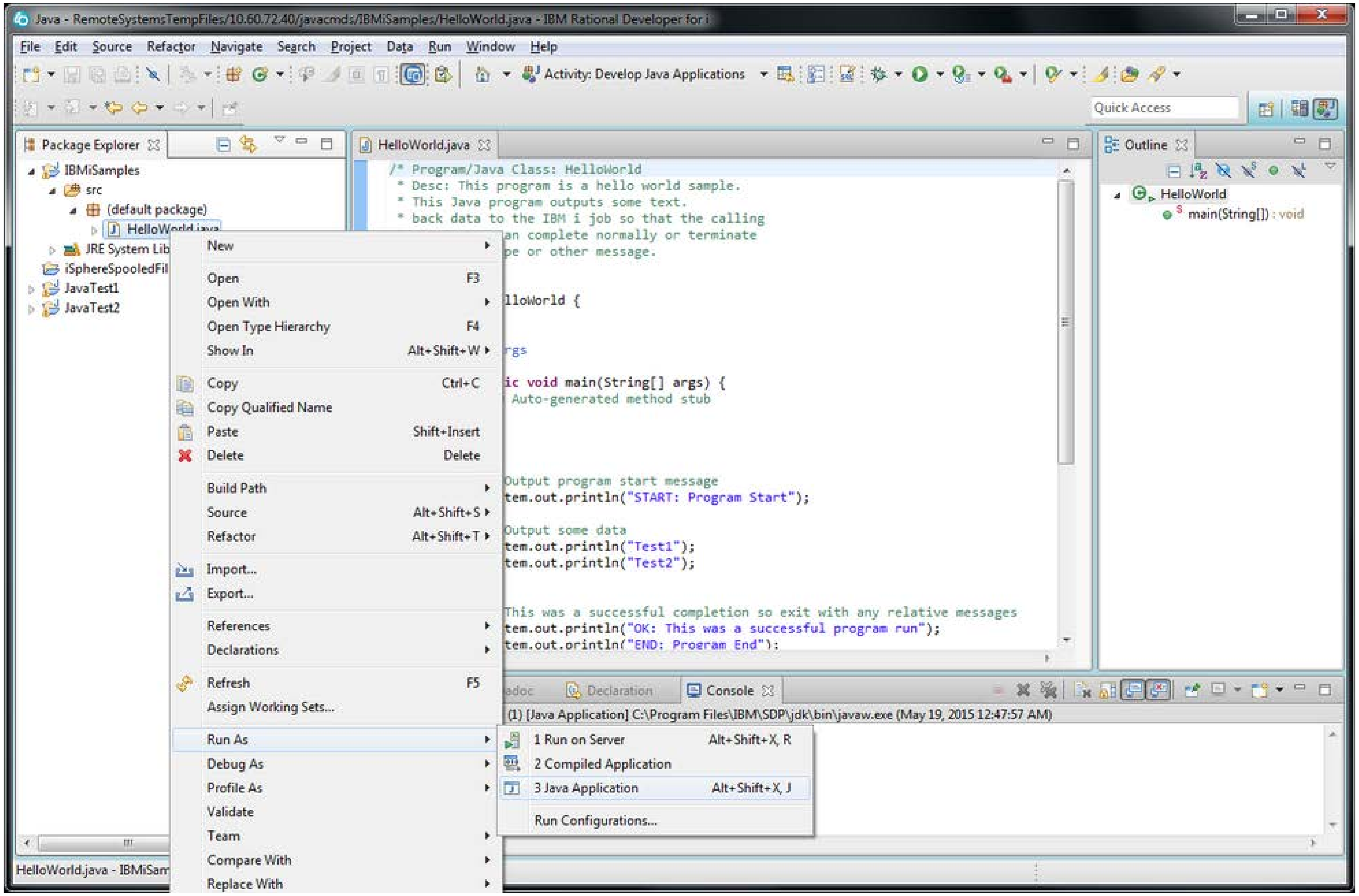
Copy/Paste from helloworld.java in IFS



```
/* Program/Java Class: HelloWorld
 * Desc: This program is a hello world sample.
 * This Java program outputs some text.
 * back data to the IBM i job so that the calling
 * CL program can complete normally or terminate
 * with an escape or other message.
 *
 */
public class HelloWorld {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub


        try {

            // Output program start message
            System.out.println("START: Program Start");

            // Output some data
```

Property	Value
Canonical Path	/javacmds/IBMiSampl
Classification	file
Extension	java
Filter string	/javacmds/IBMiSampl

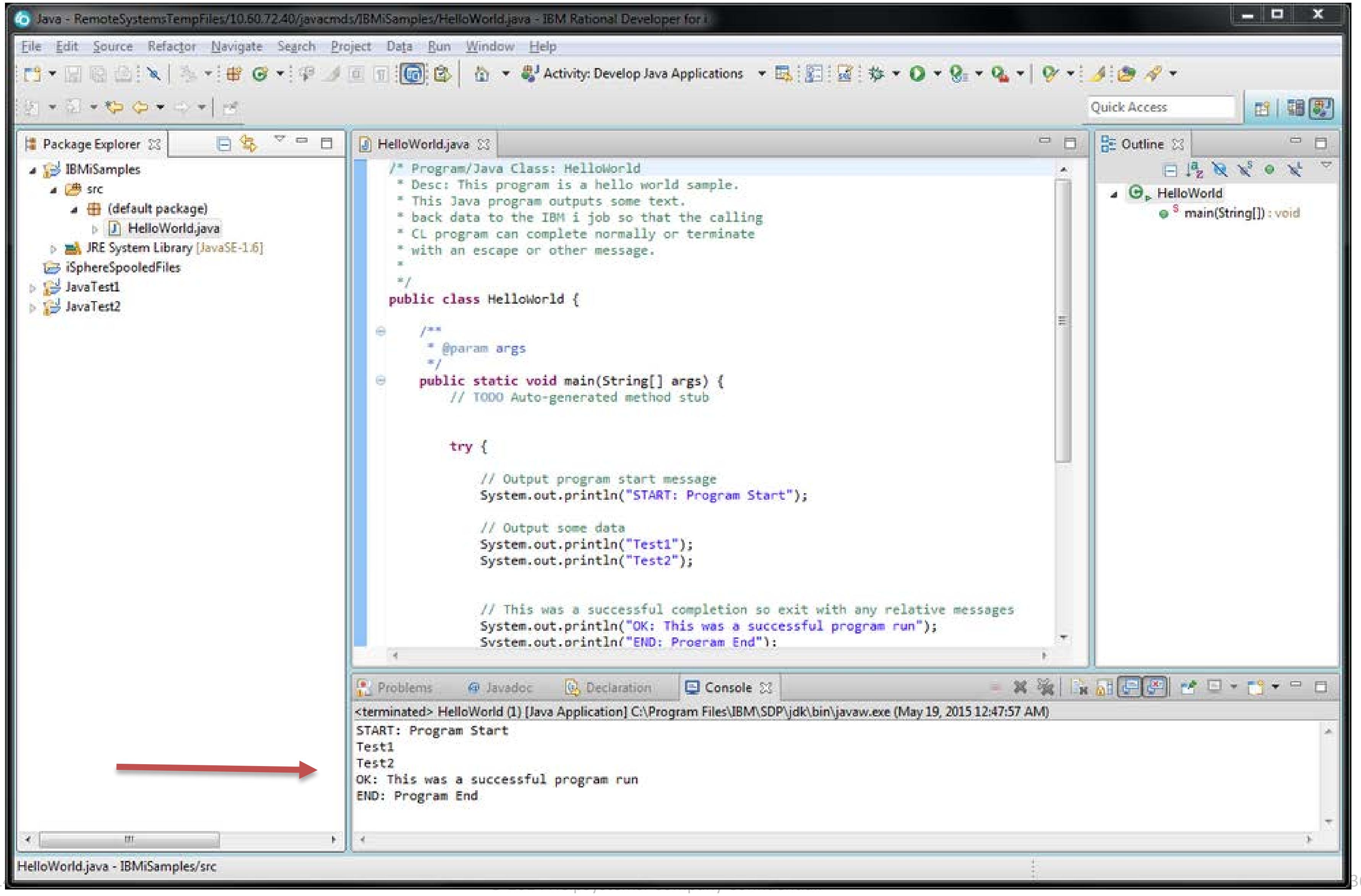
Right click and run as Java application



```
/* Program/Java Class: HelloWorld
 * Desc: This program is a hello world sample.
 * This Java program outputs some text.
 * back data to the IBM i job so that the calling
                     an complete normally or terminate
                     pe or other message.

                     lloWorld {

                     rgs

                     ic void main(String[] args) {
                      Auto-generated method stub

                     Output program start message
                     tem.out.println("START: Program Start");

                     Output some data
                     tem.out.println("Test1");
                     tem.out.println("Test2");

                     This was a successful completion so exit with any relative messages
                     tem.out.println("OK: This was a successful program run");
                     tem.out.println("END: Program End");
```

```
(1) [Java Application] C:\Program Files\IBM\SDP\jdk\bin\javaw.exe (May 19, 2015 12:47:57 AM)
```

Same output as running from green screen



Java - RemoteSystemsTempFiles/10.60.72.40/javacmds/IBMiSamples/HelloWorld.java - IBM Rational Developer for i

```
/* Program/Java Class: HelloWorld
 * Desc: This program is a hello world sample.
 * This Java program outputs some text.
 * back data to the IBM i job so that the calling
 * CL program can complete normally or terminate
 * with an escape or other message.
 *
 */
public class HelloWorld {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub


        try {

            // Output program start message
            System.out.println("START: Program Start");

            // Output some data
            System.out.println("Test1");
            System.out.println("Test2");


            // This was a successful completion so exit with any relative messages
            System.out.println("OK: This was a successful program run");
            System.out.println("END: Program End");
```

<terminated> HelloWorld (1) [Java Application] C:\Program Files\IBM\SDP\jdk\bin\javaw.exe (May 19, 2015 12:47:57 AM)

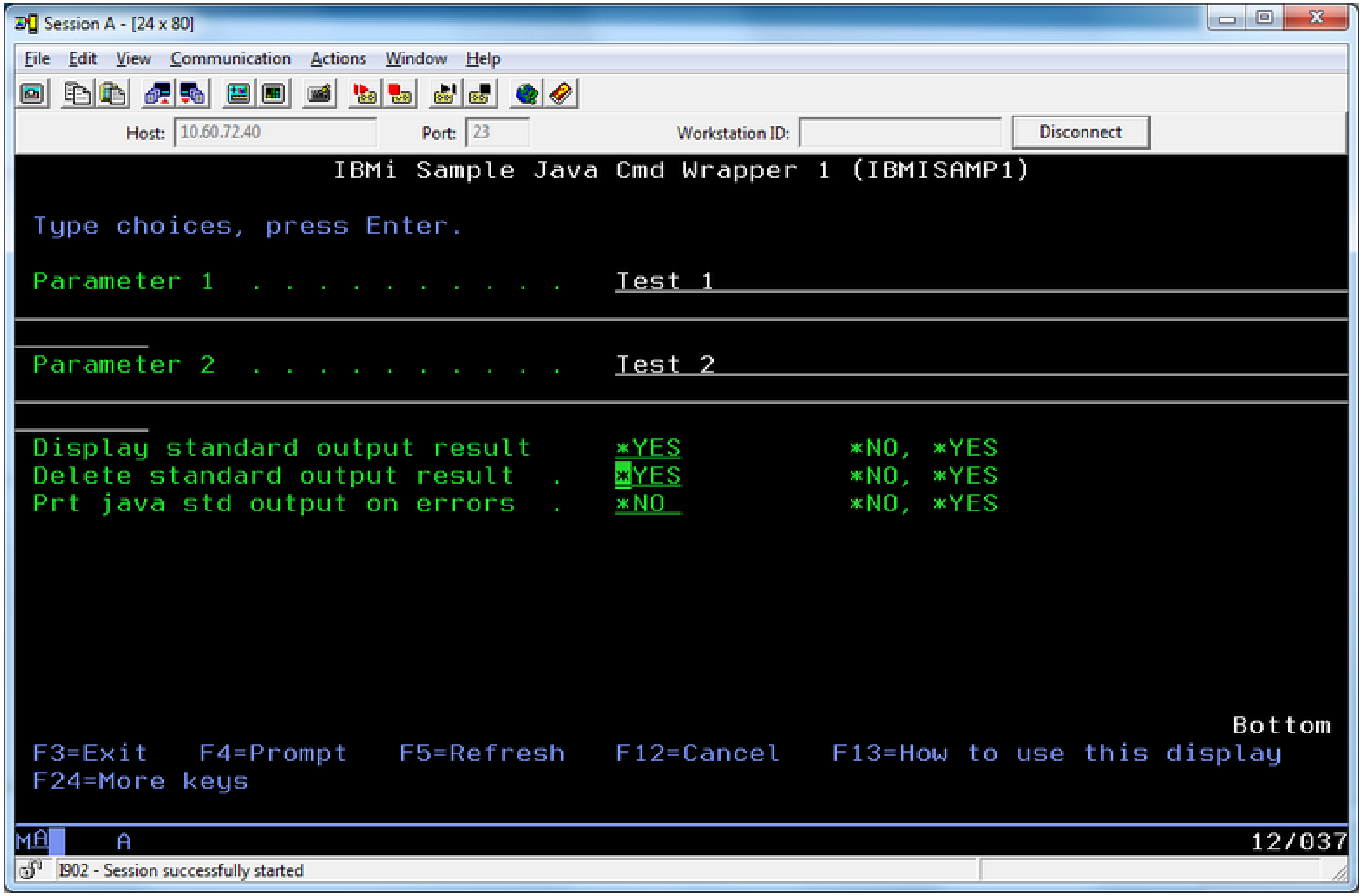
```
START: Program Start
Test1
Test2
OK: This was a successful program run
END: Program End
```

HelloWorld.java - IBMiSamples/src

Agenda

- Why use Java and CL Together
- Review development environments for Java.
- Code, compile and build
- **Work through sample commands**

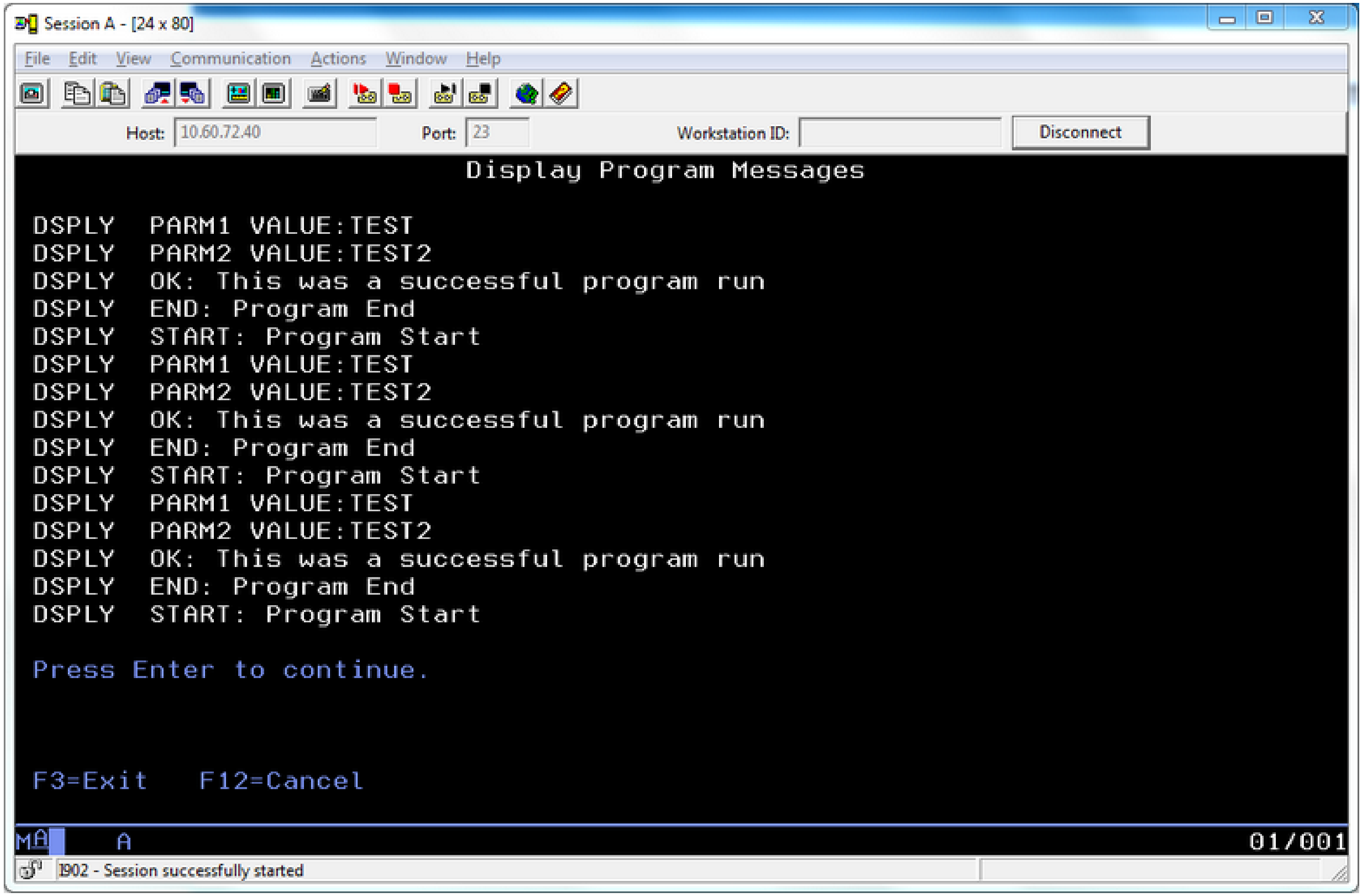
IBMISAMP1 – Call Program Return Parms



```
                  IBMi Sample Java Cmd Wrapper 1 (IBMISAMP1)

 Type choices, press Enter.

 Parameter 1 . . . . . . . . . .   Test 1


 Parameter 2 . . . . . . . . . .   Test 2


 Display standard output result    *YES          *NO, *YES
 Delete standard output result .   *YES          *NO, *YES
 Prt java std output on errors .   *NO           *NO, *YES









                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     A                                                               12/037
```

I902 - Session successfully started

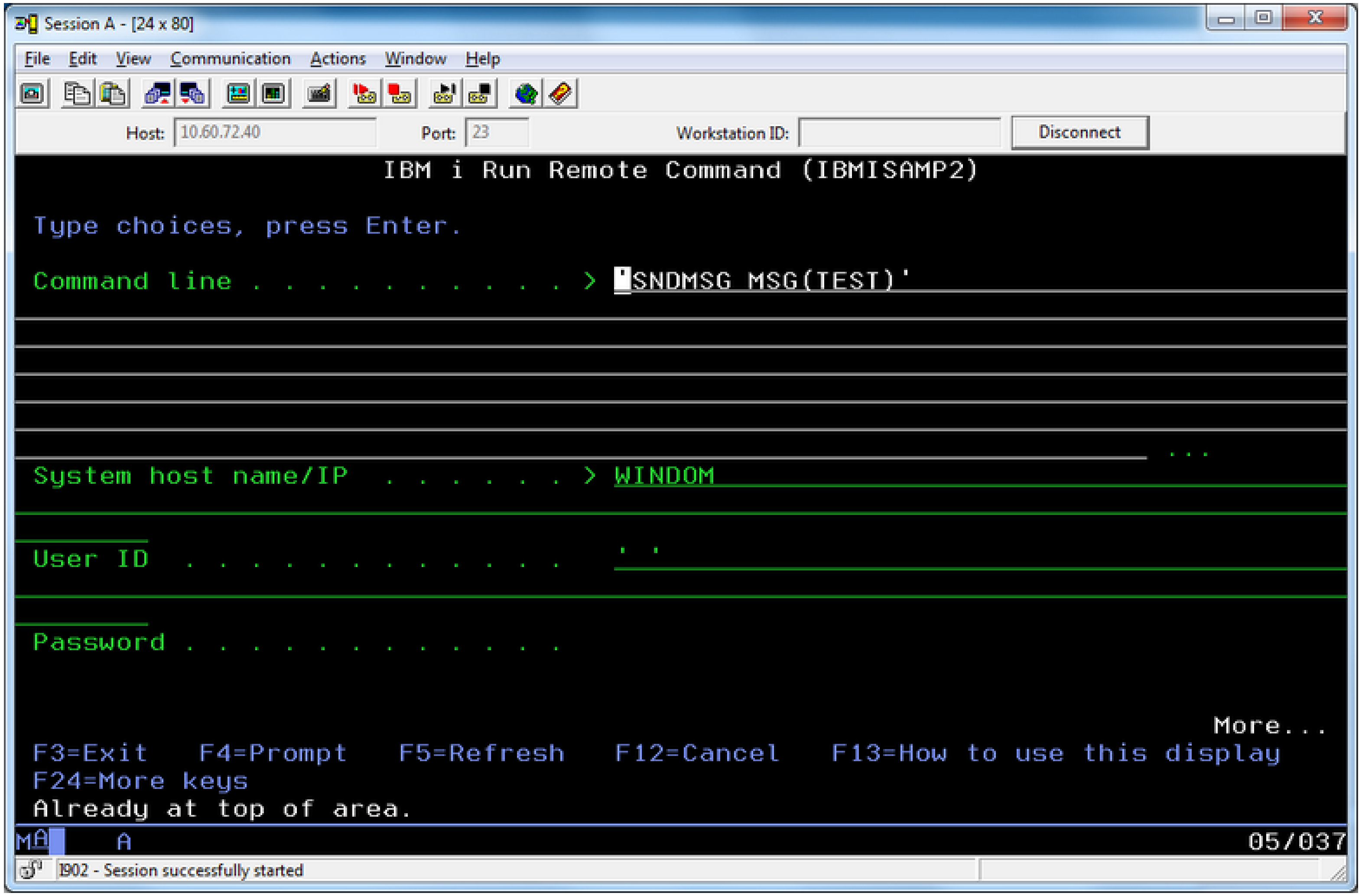
IBMISAMP1R – process STDOUT parm data



```
                         Display Program Messages

 DSPLY   PARM1 VALUE:TEST
 DSPLY   PARM2 VALUE:TEST2
 DSPLY   OK: This was a successful program run
 DSPLY   END: Program End
 DSPLY   START: Program Start
 DSPLY   PARM1 VALUE:TEST
 DSPLY   PARM2 VALUE:TEST2
 DSPLY   OK: This was a successful program run
 DSPLY   END: Program End
 DSPLY   START: Program Start
 DSPLY   PARM1 VALUE:TEST
 DSPLY   PARM2 VALUE:TEST2
 DSPLY   OK: This was a successful program run
 DSPLY   END: Program End
 DSPLY   START: Program Start

 Press Enter to continue.



 F3=Exit   F12=Cancel
MA     A                                                               01/001
```

I902 - Session successfully started

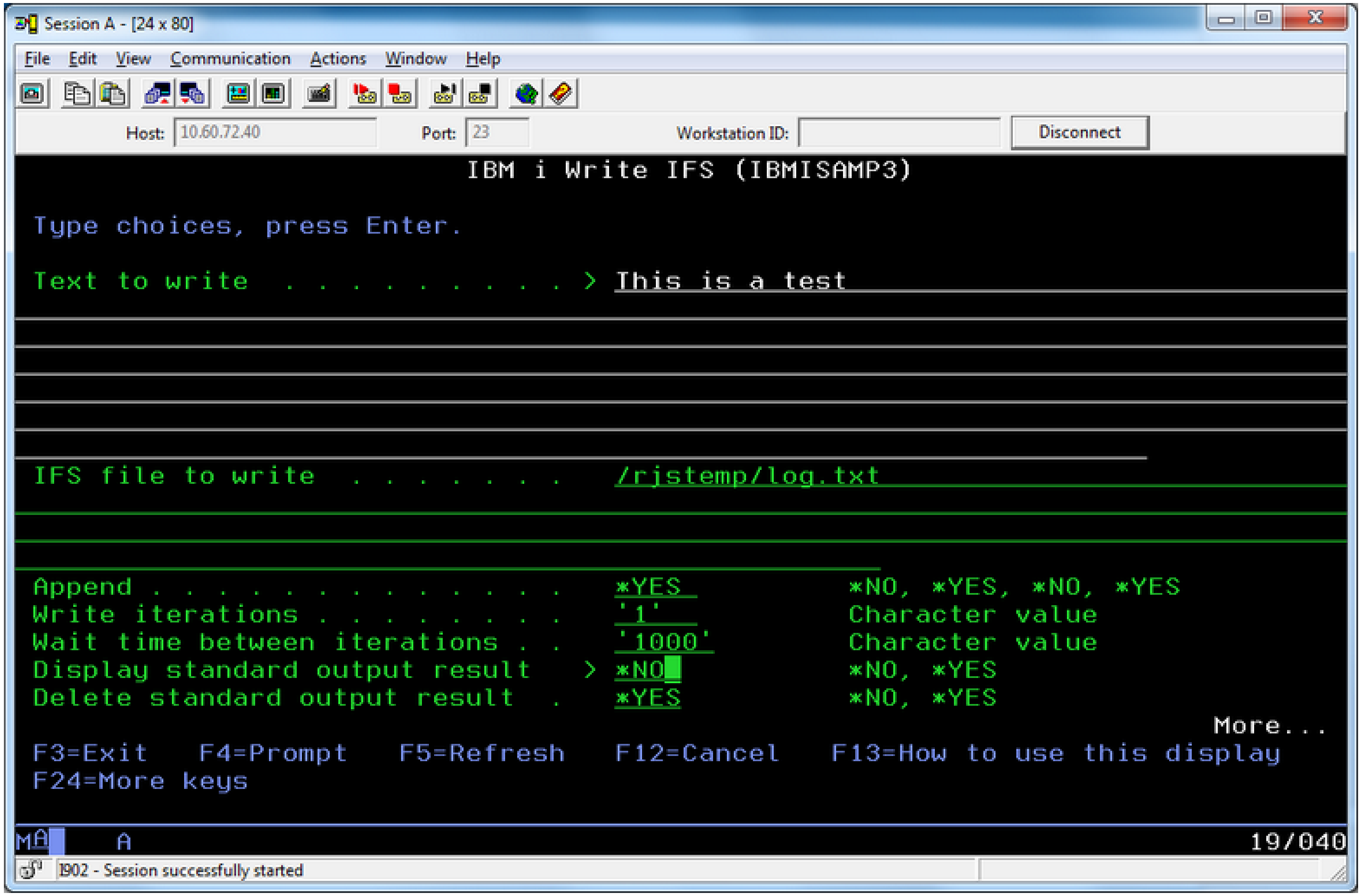
IBMISAMP2 – Run Remote System Command



```
                    IBM i Run Remote Command (IBMISAMP2)

 Type choices, press Enter.

 Command line . . . . . . . . . . > 'SNDMSG MSG(TEST)'



                                                                     ...
 System host name/IP . . . . . . > WINDOM

 User ID . . . . . . . . . . . .   ' '

 Password . . . . . . . . . . . .

                                                                    More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Already at top of area.
MA     A                                                             05/037
```

I902 - Session successfully started

IBMISAMP3 – Write to IFS File or Log


Session A - [24 x 80]
File Edit View Communication Actions Window Help
Host: 10.60.72.40 Port: 23 Workstation ID: Disconnect

```
                              IBM i Write IFS (IBMISAMP3)

 Type choices, press Enter.

 Text to write . . . . . . . . . > This is a test





 IFS file to write . . . . . . .   /rjstemp/log.txt


 Append . . . . . . . . . . . . .   *YES          *NO, *YES, *NO, *YES
 Write iterations . . . . . . . .   '1'           Character value
 Wait time between iterations . .   '1000'        Character value
 Display standard output result  > *NO           *NO, *YES
 Delete standard output result  .   *YES          *NO, *YES
                                                                      More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     A                                                                 19/040
```

I902 - Session successfully started

IBMPUTFILE – Send IFS File to Windows Share

- This CL command sends a binary file from the IFS to a Windows share using the JCIFS java API

- IBMPUTFILE DOMAIN(RJSTESTVM01)

  ```
  SERVER('rjstestvm01.rjsintranet.com')
  SHARENAME('delivernow')
  USER('rjs')
  PASSWORD(‘password’)
  INPUTFILE('/rjstemp/test.txt')
  OUTPUTFILE('/test.txt')
  REPLACE(*YES)
  DSPSTDOUT(*YES)
  ```

IBMGETFILE – Get IFS File from Windows Share

- This CL command reads a binary file from a Windows share using the JCIFS java API

-
  ```
  IBMGETFILE DOMAIN(RJSTESTVM01)
      SERVER('rjstestvm01.rjsintranet.com')
      SHARENAME('delivernow')
      USER('rjs')
      PASSWORD(‘password’)
      INPUTFILE('/test.txt')
      OUTPUTFILE('/rjstemp/test2.txt')
      DSPSTDOUT(*YES)
  ```

IBMSETUSER – Set Netbios User and Password

- This CL command sets a global user id and password for netbios commands and stores in data area NBUSER and NBPASS. If *GLOBAL is specified for user and password on IBMGETFILE and IBMPUTFILE, these values are used. This is nice for storing a global netbios login.

- IBMSETUSER OPTION(*SET)
```
IBMSETUSER OPTION(*SET)
    NBUSER('user')
    NBPASS('password')
```

What commands would you like to see ?

- Send me feedback on additional commands I can add to this library.
- What would you like to see ?

Wrap Up

- **Why use Java and CL Together**
- Review development environments for Java.
- Code, compile and build
- Work through sample commands
- Download examples, pay and provide feedback

Flexible IT Software Solutions

Our solutions help customers save time, money and eliminate errors. And each functional area offers these benefits:

Systems & Network Management	Business Intelligence	Security & Compliance
• Automate manual IT and business processes • Remove costly scheduling gaps and delays • Meet Service Level Agreements • Avoid outages and network slowdowns	• Access real-time data from anywhere • Run lightning-fast large queries • View dashboards from any browser • Present metrics in a user-friendly format	• Protect critical data at every access level • Centralize user profile administration • Eliminate viruses before they spread • Stay informed when security events occur • Simplify compliance and pass any audit

Solutions Portfolio

helpsystems

Systems & Network Management

robot
- Robot SCHEDULE
- Robot SCHEDULE Enterprise
- Robot CONSOLE
- Robot NETWORK
- Robot ALERT
- Robot REPORTS
- Robot SAVE
- Robot SPACE
- Robot REPLAY

skybot
- Skybot Scheduler

BYTWARE
- StandGuard Anti-Virus
- StandGuard Recycle Bin
- MessengerPlus
- MessengerConsole
- PeekPlus

rjs Software Systems
- SignHere
- iForms
- WebDocs
- WebForms
- DeliverNow

InterMapper
- InterMapper
- InterMapper RemoteAccess
- InterMapper Flows
- Splunk App

AutoMate
- AutoMate
- AutoMate BPA Server

CCSS
- QSystem Monitor
- QMessage Monitor
- QRemote Control

halcyon
- Enterprise Console
- MQ Manager
- Performance Analyzer
- Disk Space Manager
- Spooled File Manager
- Restricted Tasks Manager
- HA-MX Monitor
- Message Communicator
- Network Server Suite

Business Intelligence

SEQUEL
- SEQUEL
- SEQUEL Web Interface
- SEQUEL Data Warehouse
- ESEND

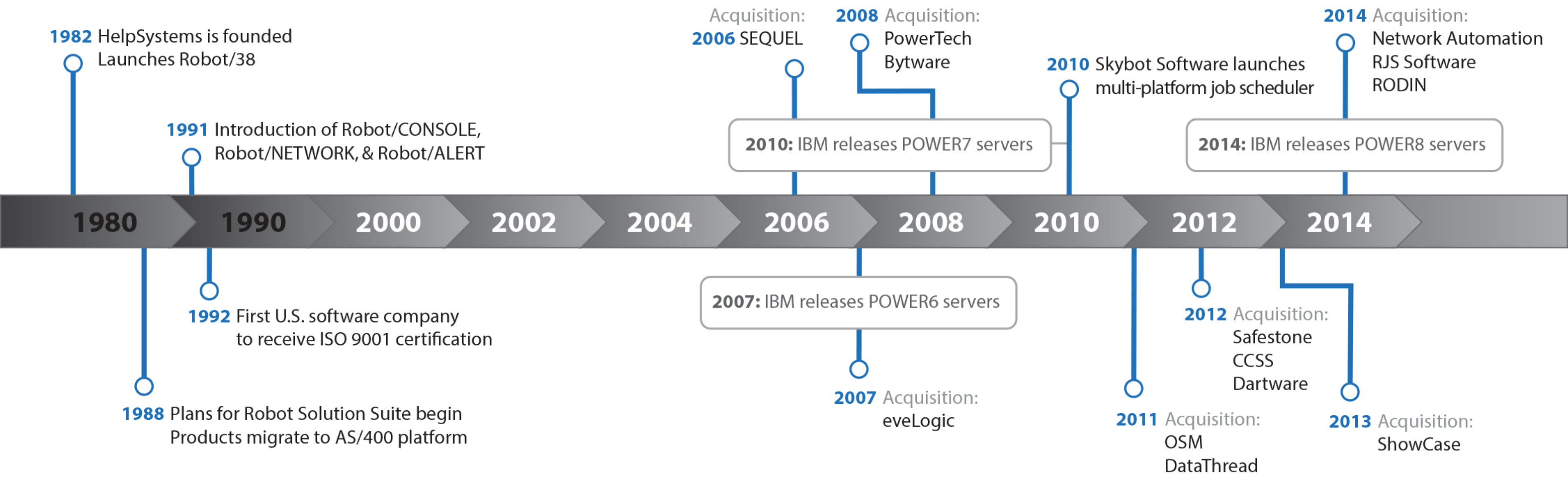
ShowCase
- Query & Report Writer
- Warehouse Manager
- Warehouse Builder

Security & Compliance

PowerTech
- Network Security
- Compliance Monitor
- Authority Broker
- Interact
- DataThread
- Command Security
- PowerAdmin

safestone
- Compliance Center
- Multiple Systems Administrator
- Network Traffic Controller
- Powerful User Passport
- User Profile Manager
- Password Self Help
- Agent for RSA SecurID
- iConnect

30 Years of Growth And Innovation


1982 HelpSystems is founded
Launches Robot/38
1991 Introduction of Robot/CONSOLE, Robot/NETWORK, & Robot/ALERT
Acquisition: 2006 SEQUEL
2008 Acquisition: PowerTech Bytware
2010: IBM releases POWER7 servers
2010 Skybot Software launches multi-platform job scheduler
2014 Acquisition: Network Automation RJS Software RODIN
2014: IBM releases POWER8 servers
1980
1990
2000
2002
2004
2006
2008
2010
2012
2014
1988 Plans for Robot Solution Suite begin
Products migrate to AS/400 platform
1992 First U.S. software company to receive ISO 9001 certification
2007: IBM releases POWER6 servers
2007 Acquisition: eveLogic
2011 Acquisition: OSM DataThread
2012 Acquisition: Safestone CCSS Dartware
2013 Acquisition: ShowCase

Document Lifecycle Management

Capture → Manage → Distribute → Secure

SignHere	Electronic signature capture
iForms	Electronic forms & reporting
WebDocs	Document management
WebForms	Web-based forms capture
DeliverNow	Automated report distribution
Smart AP	Automated invoice processing

Thank you for attending today !

Contact Info

Website:

www.helpsystems.com/rjs

Telephone:

800-328-1000 *sales*

+1 952-933-0609 *support*

Presenter

Richard Schoen
richard.schoen@helpsystems.com
Phone: 952-486-6802