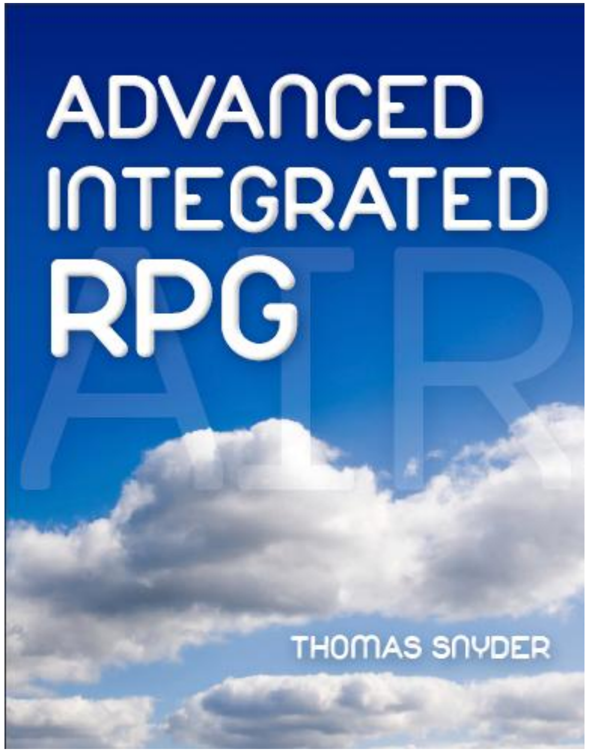
Advanced Integrated RPG

Integrating RPG with Java and Open Source



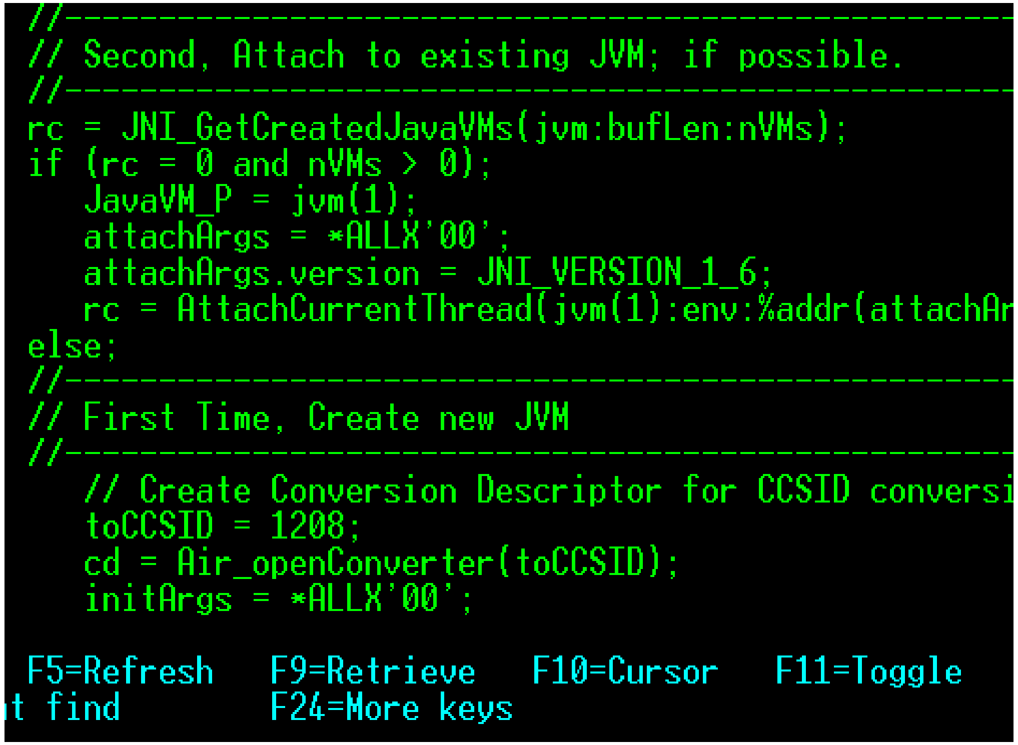
Session 1

Using Java with RPG

Tom Snyder

RPG, OPM and ILE

- Free Formatted RPG
- Activation Groups
- Procedures
- Service Programs



```
//----------------------------------------------------------
// Second, Attach to existing JVM; if possible.
//----------------------------------------------------------
rc = JNI_GetCreatedJavaVMs(jvm:bufLen:nVMs);
if (rc = 0 and nVMs > 0);
   JavaVM_P = jvm(1);
   attachArgs = *ALLX'00';
   attachArgs.version = JNI_VERSION_1_6;
   rc = AttachCurrentThread(jvm(1):env:%addr(attachAr
else;
//----------------------------------------------------------
// First Time, Create new JVM
//----------------------------------------------------------
   // Create Conversion Descriptor for CCSID conversi
   toCCSID = 1208;
   cd = Air_openConverter(toCCSID);
   initArgs = *ALLX'00';

 F5=Refresh    F9=Retrieve    F10=Cursor    F11=Toggle
t find         F24=More keys
```

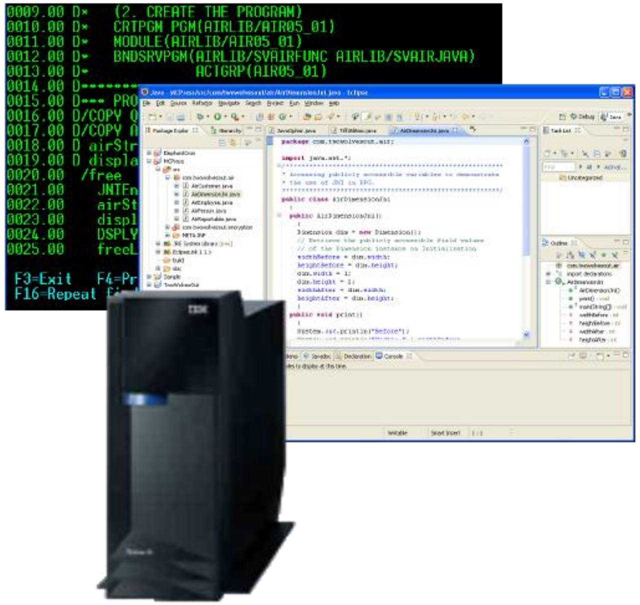
Java

- Cross Platform
- Virtual Machine
- Supported on the IBM i
- Access to Open Source

Open Source

- Apache POI – The Java API for Microsoft Documents
- iText – Java-PDF Library
- JavaMail – Platform Independent Email Framework

Using Java with RPG

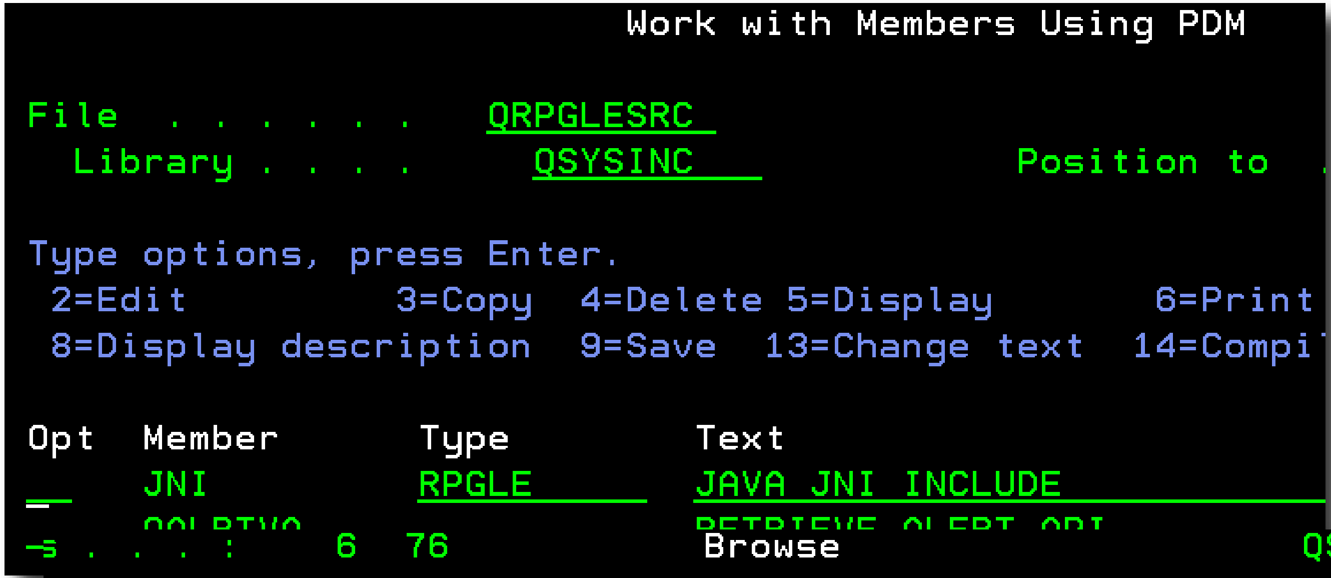


```
0009.00 D*    (2. CREATE THE PROGRAM)
0010.00 D*    CRTPGM PGM(AIRLIB/AIR05_01)
0011.00 D*    MODULE(AIRLIB/AIR05_01)
0012.00 D*    BNDSRVPGM(AIRLIB/SVAIRFUNC AIRLIB/SVAIRJAVA)
0013.00 D*                ACTGRP(AIR05_01)
0014.00 D*******
0015.00 D*** PRO
0016.00 D/COPY Q
0017.00 D/COPY A
0018.00 D airStr
0019.00 D displa
0020.00  /free
0021.00     JNIEn
0022.00     airSt
0023.00     displ
0024.00     DSPLY
0025.00     freeL

 F3=Exit   F4=Pr
 F16=Repeat fi
```

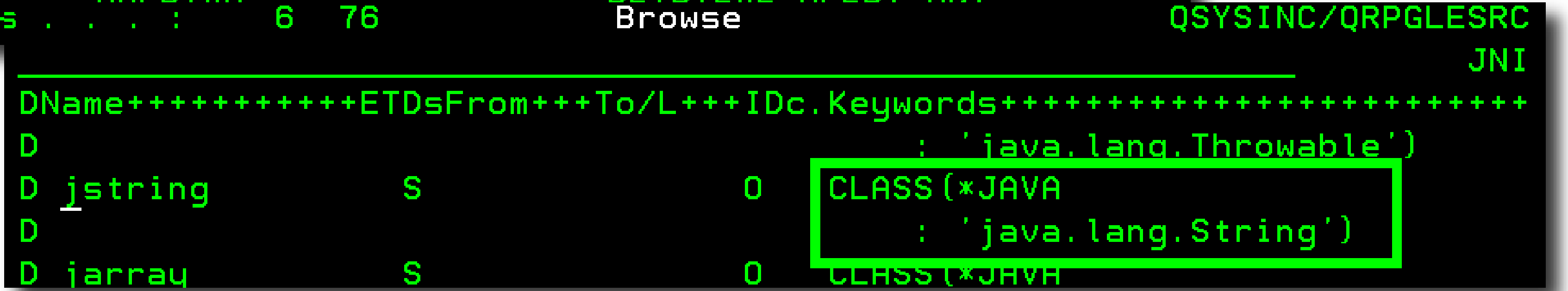
```
package com.twoweolvesout.air;

import java.awt.*;
/*****************************************************
 * Accessing publicly accessible variables to demonstrate
 * the use of JNI in RPG.
 *****************************************************/
public class AirDimensionJni
{
  public AirDimensionJni()
  {
    Dimension dim = new Dimension();
    // Retrieve the publicly accessible field values
    // of the Dimension instance on Initialization
    widthBefore = dim.width;
    heightBefore = dim.height;
    dim.width = 1;
    dim.height = 1;
    widthAfter = dim.width;
    heightAfter = dim.height;
  }
  public void print()
  {
    System.out.println("Before");
```

- Accessing Java Objects from RPG
- Working with the Java Virtual Machine
- Java Native Interface

Making a Reference Variable

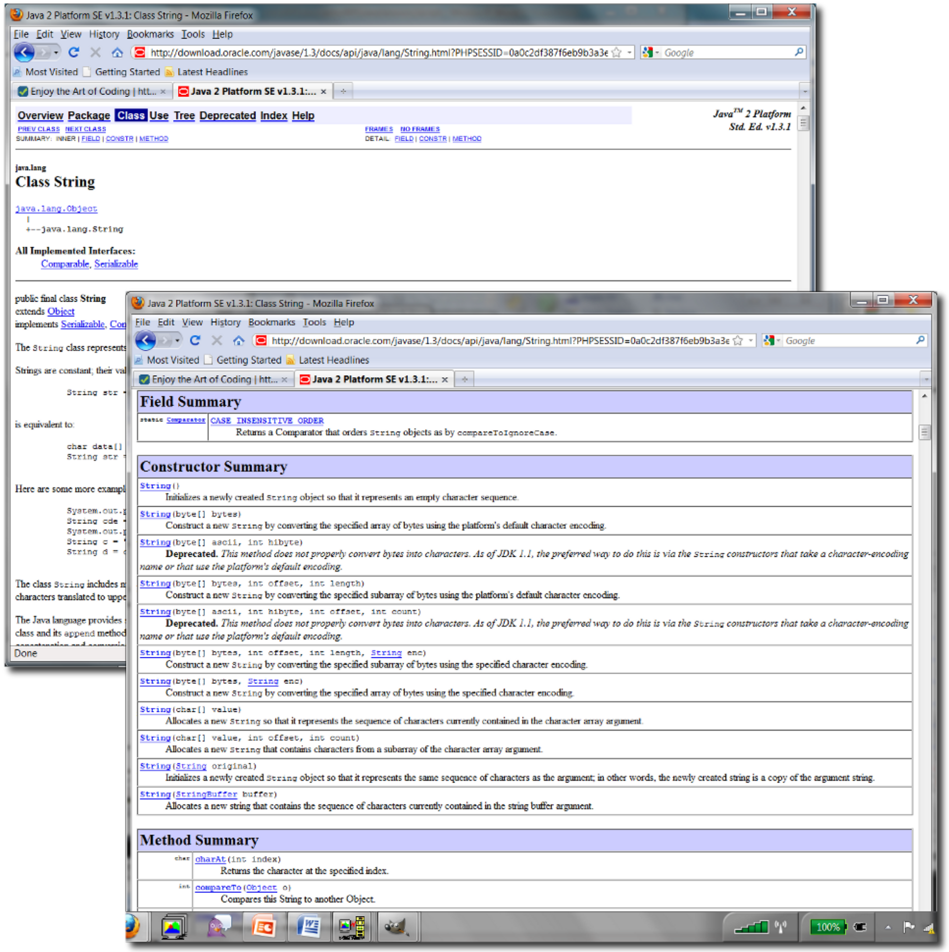


```
                        Work with Members Using PDM

File  . . . . . .   QRPGLESRC
  Library . . . .     QSYSINC                 Position to  .

Type options, press Enter.
 2=Edit          3=Copy  4=Delete 5=Display       6=Print
 8=Display description  9=Save  13=Change text  14=Compi

Opt  Member       Type        Text
     JNI          RPGLE       JAVA JNI INCLUDE
```



```
 s . . . :      6   76              Browse                     QSYSINC/QRPGLESRC
                                                                             JNI
DName+++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D                                          : 'java.lang.Throwable')
D jstring         S              O         CLASS(*JAVA
D                                          : 'java.lang.String')
D jarray          S              O         CLASS(*JAVA
```

JavaDocs


Java 2 Platform SE v1.3.1: Class String - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://download.oracle.com/javase/1.3/docs/api/java/lang/String.html?PHPSESSID=0a0c2df387f6eb9b3a3e
Google
Most Visited
Getting Started
Latest Headlines
Enjoy the Art of Coding | htt...
Java 2 Platform SE v1.3.1:...
Overview Package Class Use Tree Deprecated Index Help
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: INNER | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
Done
100%

Java™ 2 Platform
Std. Ed. v1.3.1

java.lang
Class String

```
java.lang.Object
  |
  +--java.lang.String
```

All Implemented Interfaces:
Comparable, Serializable

public final class **String**
extends Object
implements Serializable, Com

The `String` class represents

Strings are constant; their val

```
        String str
```

is equivalent to:

```
        char data[]
        String str
```

Here are some more exampl

```
        System.out.
        String cde
        System.out.
        String c =
        String d =
```

The class `String` includes m
characters translated to uppe

The Java language provides
class and its `append` method

Field Summary

static Comparator	`CASE_INSENSITIVE_ORDER` Returns a Comparator that orders `String` objects as by `compareToIgnoreCase`.

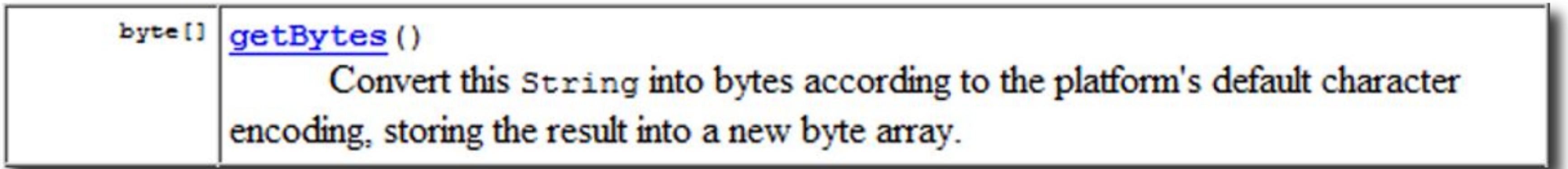
Constructor Summary

`String()` Initializes a newly created `String` object so that it represents an empty character sequence.
`String(byte[] bytes)` Construct a new `String` by converting the specified array of bytes using the platform's default character encoding.
`String(byte[] ascii, int hibyte)` **Deprecated.** *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the `String` constructors that take a character-encoding name or that use the platform's default encoding.*
`String(byte[] bytes, int offset, int length)` Construct a new `String` by converting the specified subarray of bytes using the platform's default character encoding.
`String(byte[] ascii, int hibyte, int offset, int count)` **Deprecated.** *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the `String` constructors that take a character-encoding name or that use the platform's default encoding.*
`String(byte[] bytes, int offset, int length, String enc)` Construct a new `String` by converting the specified subarray of bytes using the specified character encoding.
`String(byte[] bytes, String enc)` Construct a new `String` by converting the specified array of bytes using the specified character encoding.
`String(char[] value)` Allocates a new `String` so that it represents the sequence of characters currently contained in the character array argument.
`String(char[] value, int offset, int count)` Allocates a new `String` that contains characters from a subarray of the character array argument.
`String(String original)` Initializes a newly created `String` object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
`String(StringBuffer buffer)` Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.

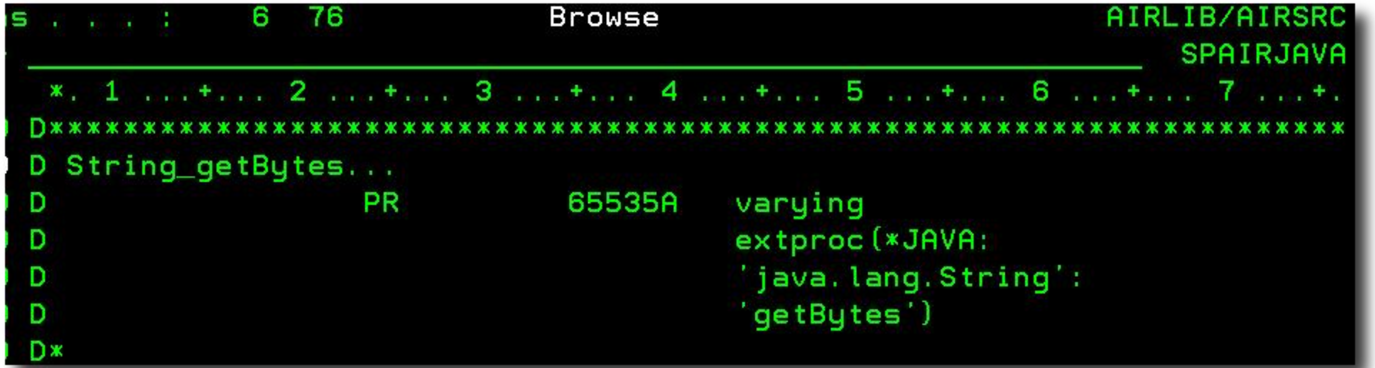
Method Summary

char	`charAt(int index)` Returns the character at the specified index.
int	`compareTo(Object o)` Compares this String to another Object.

String Class, getBytes Method

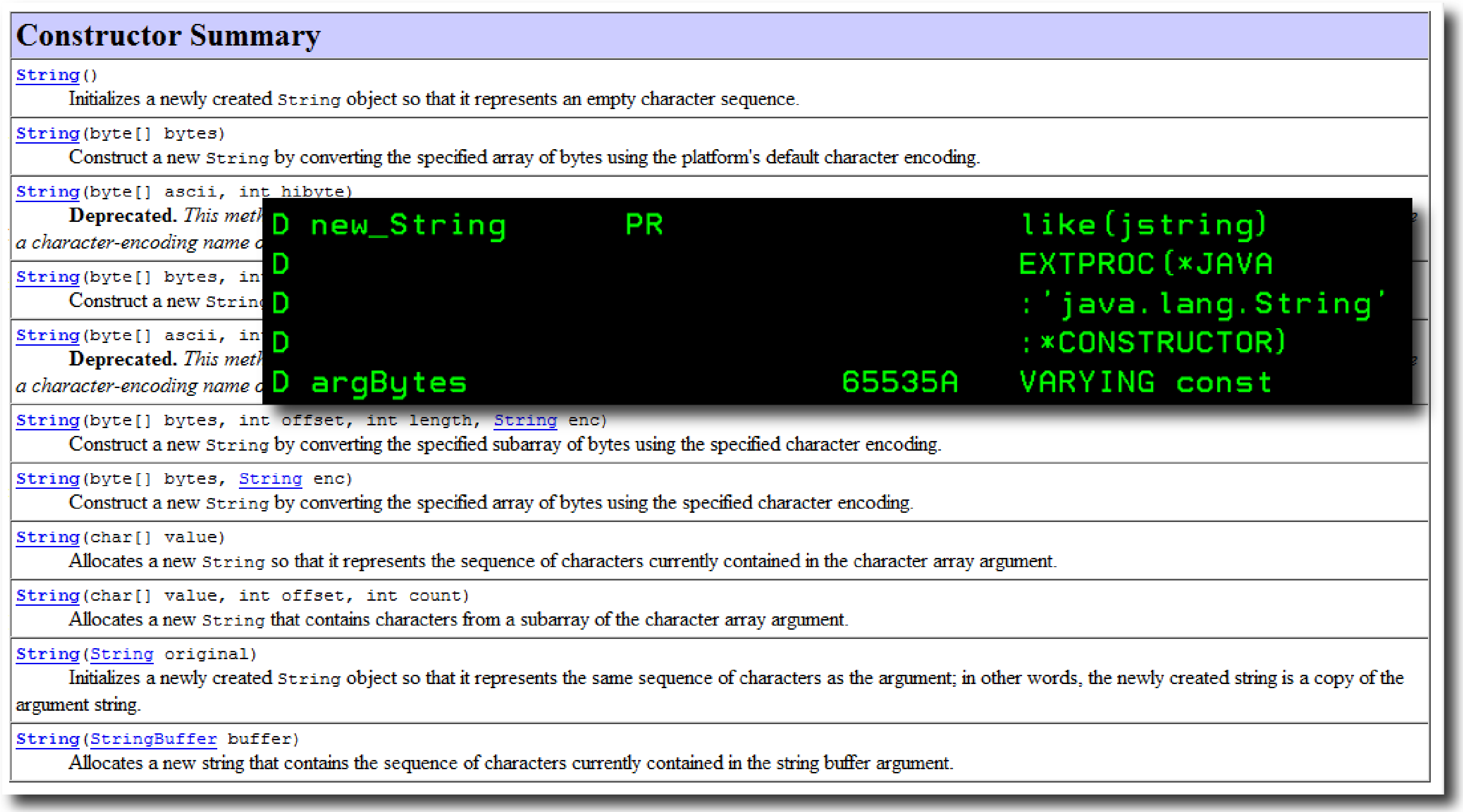


byte[]	getBytes() Convert this String into bytes according to the platform's default character encoding, storing the result into a new byte array.



```
s . . . :    6  76           Browse                    AIRLIB/AIRSRC
                                                           SPAIRJAVA
  *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+.
 D*************************************************************************
 D String_getBytes...
 D                PR          65535A   varying
 D                                     extproc(*JAVA:
 D                                     'java.lang.String':
 D                                     'getBytes')
 D*
```

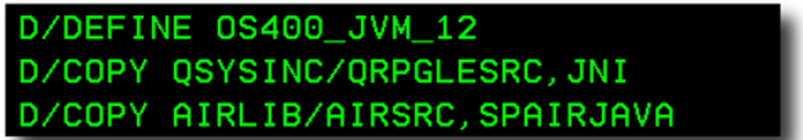
String Class Constructor



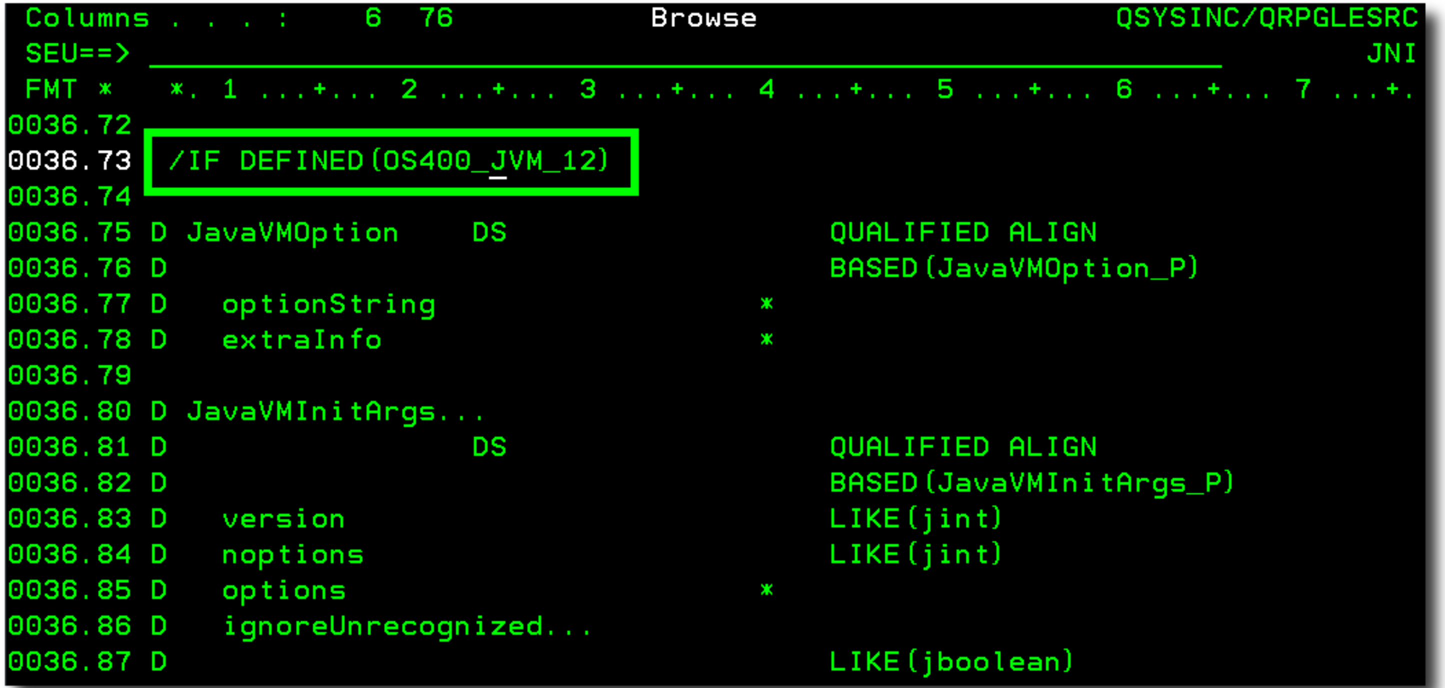
Constructor Summary
String()   Initializes a newly created String object so that it represents an empty character sequence.
String(byte[] bytes)   Construct a new String by converting the specified array of bytes using the platform's default character encoding.
String(byte[] ascii, int hibyte)   **Deprecated.** *This meth*   *a character-encoding name o*
String(byte[] bytes, in   Construct a new Strin
String(byte[] ascii, in   **Deprecated.** *This meth*   *a character-encoding name o*
String(byte[] bytes, int offset, int length, String enc)   Construct a new String by converting the specified subarray of bytes using the specified character encoding.
String(byte[] bytes, String enc)   Construct a new String by converting the specified array of bytes using the specified character encoding.
String(char[] value)   Allocates a new String so that it represents the sequence of characters currently contained in the character array argument.
String(char[] value, int offset, int count)   Allocates a new String that contains characters from a subarray of the character array argument.
String(String original)   Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
String(StringBuffer buffer)   Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.

```
D new_String      PR                  like(jstring)
D                                     EXTPROC(*JAVA
D                                     :'java.lang.String'
D                                     :*CONSTRUCTOR)
D argBytes                65535A      VARYING const
```

QSYSINC/QRPGLESRC, JNI

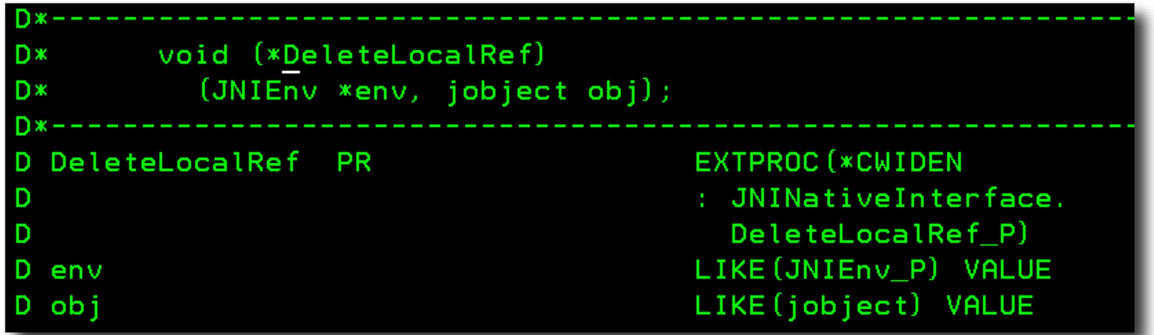


```
D/DEFINE OS400_JVM_12
D/COPY QSYSINC/QRPGLESRC,JNI
D/COPY AIRLIB/AIRSRC,SPAIRJAVA
```

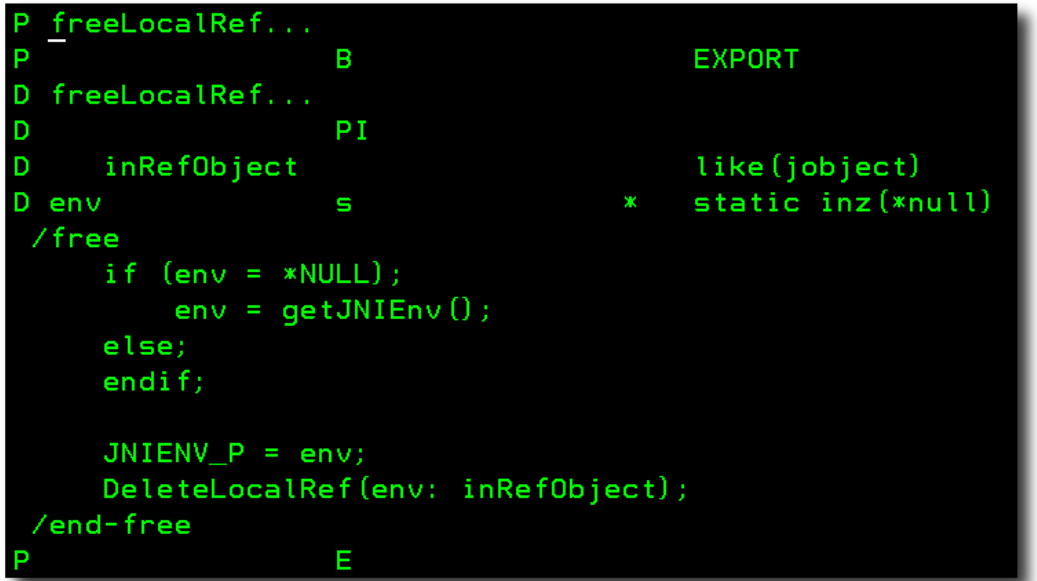


```
 Columns . . . :    6  76           Browse                    QSYSINC/QRPGLESRC
 SEU==> ______________________________________________________________      JNI
 FMT *  *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+.
0036.72
0036.73  /IF DEFINED(OS400_JVM_12)
0036.74
0036.75 D JavaVMOption    DS                    QUALIFIED ALIGN
0036.76 D                                       BASED(JavaVMOption_P)
0036.77 D   optionString                 *
0036.78 D   extraInfo                    *
0036.79
0036.80 D JavaVMInitArgs...
0036.81 D                 DS                    QUALIFIED ALIGN
0036.82 D                                       BASED(JavaVMInitArgs_P)
0036.83 D   version                             LIKE(jint)
0036.84 D   noptions                            LIKE(jint)
0036.85 D   options                      *
0036.86 D   ignoreUnrecognized...
0036.87 D                                       LIKE(jboolean)
```

Garbage Collection

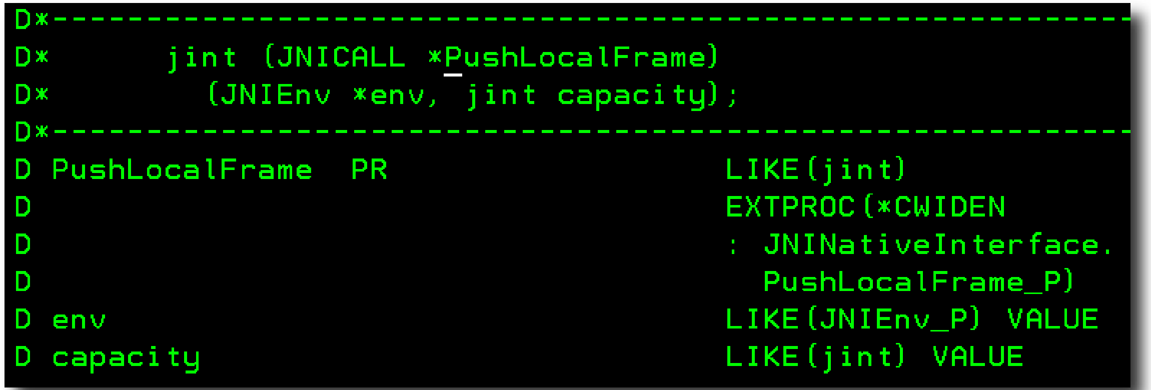


```
D*---------------------------------------------------------------
D*      void (*DeleteLocalRef)
D*        (JNIEnv *env, jobject obj);
D*---------------------------------------------------------------
D DeleteLocalRef  PR                    EXTPROC(*CWIDEN
D                                       : JNINativeInterface.
D                                         DeleteLocalRef_P)
D env                                   LIKE(JNIEnv_P) VALUE
D obj                                   LIKE(jobject) VALUE
```

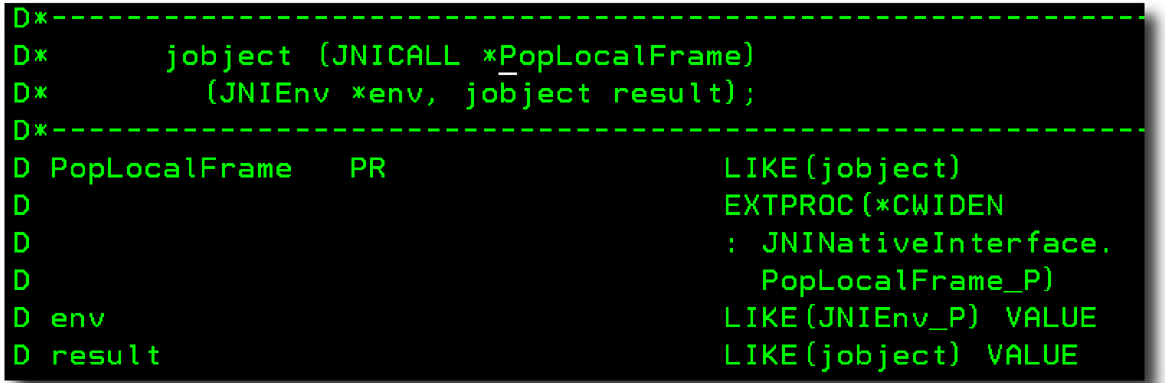


```
P freeLocalRef...
P                 B                   EXPORT
D freeLocalRef...
D                 PI
D    inRefObject                      like(jobject)
D env             s               *   static inz(*null)
 /free
     if (env = *NULL);
         env = getJNIEnv();
     else;
     endif;

     JNIENV_P = env;
     DeleteLocalRef(env: inRefObject);
 /end-free
P                 E
```

Pushing And Popping Frames

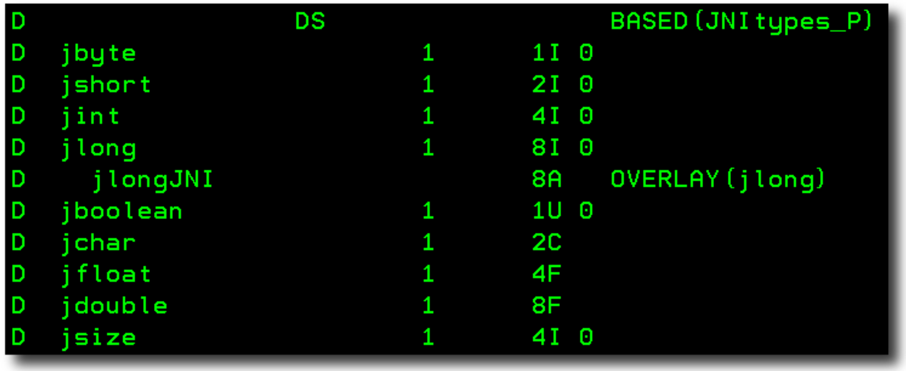


```
D*----------------------------------------------------------
D*      jint (JNICALL *PushLocalFrame)
D*        (JNIEnv *env, jint capacity);
D*----------------------------------------------------------
D PushLocalFrame  PR                  LIKE(jint)
D                                     EXTPROC(*CWIDEN
D                                     : JNINativeInterface.
D                                       PushLocalFrame_P)
D env                                 LIKE(JNIEnv_P) VALUE
D capacity                            LIKE(jint) VALUE
```

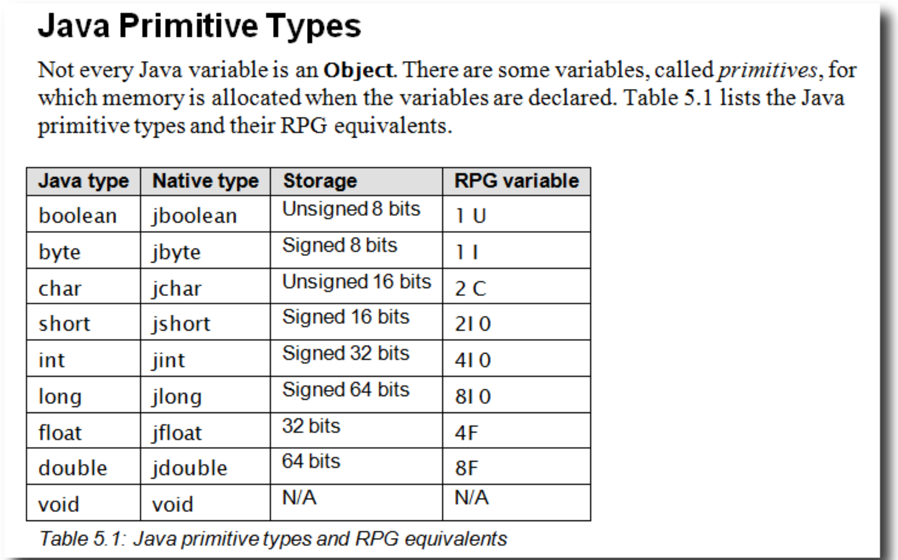


```
D*----------------------------------------------------------
D*      jobject (JNICALL *PopLocalFrame)
D*        (JNIEnv *env, jobject result);
D*----------------------------------------------------------
D PopLocalFrame   PR                  LIKE(jobject)
D                                     EXTPROC(*CWIDEN
D                                     : JNINativeInterface.
D                                       PopLocalFrame_P)
D env                                 LIKE(JNIEnv_P) VALUE
D result                              LIKE(jobject) VALUE
```

Java Primitive Types



```
D                  DS                     BASED(JNItypes_P)
D  jbyte                   1      1I 0
D  jshort                  1      2I 0
D  jint                    1      4I 0
D  jlong                   1      8I 0
D    jlongJNI                     8A      OVERLAY(jlong)
D  jboolean                1      1U 0
D  jchar                   1      2C
D  jfloat                  1      4F
D  jdouble                 1      8F
D  jsize                   1      4I 0
```



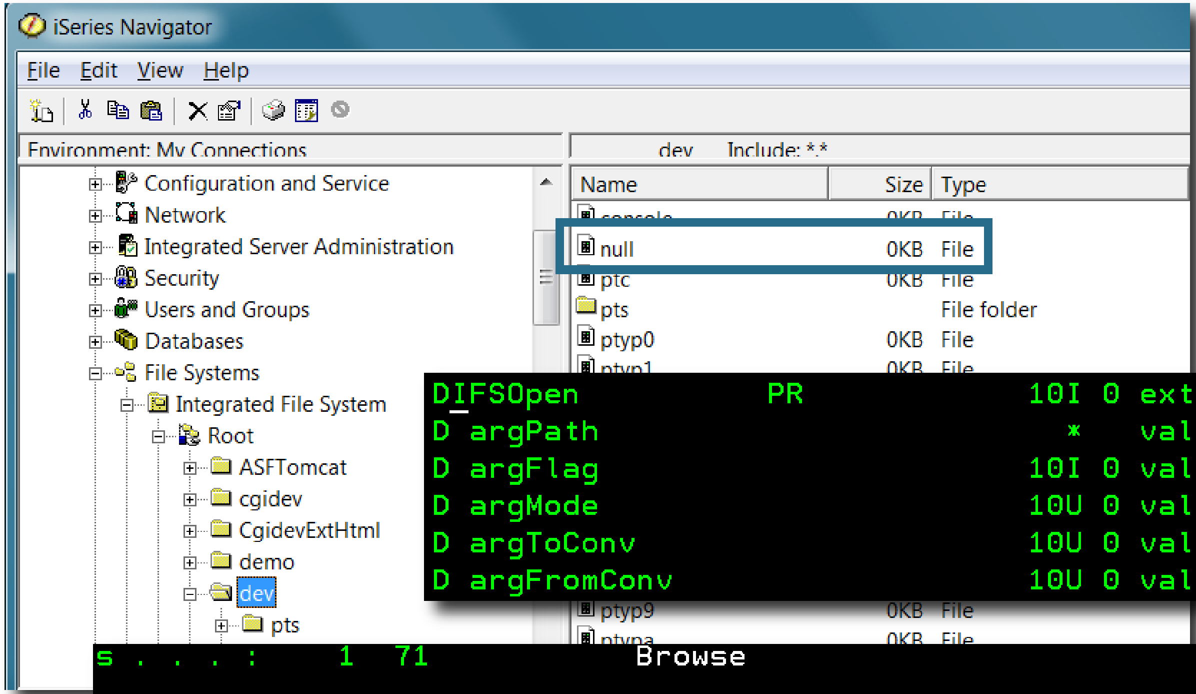
Java Primitive Types

Not every Java variable is an **Object**. There are some variables, called *primitives*, for which memory is allocated when the variables are declared. Table 5.1 lists the Java primitive types and their RPG equivalents.

Java type	Native type	Storage	RPG variable
boolean	jboolean	Unsigned 8 bits	1 U
byte	jbyte	Signed 8 bits	1 I
char	jchar	Unsigned 16 bits	2 C
short	jshort	Signed 16 bits	2I 0
int	jint	Signed 32 bits	4I 0
long	jlong	Signed 64 bits	8I 0
float	jfloat	32 bits	4F
double	jdouble	64 bits	8F
void	void	N/A	N/A

Table 5.1: Java primitive types and RPG equivalents

STDIN, STDOUT, STDERR



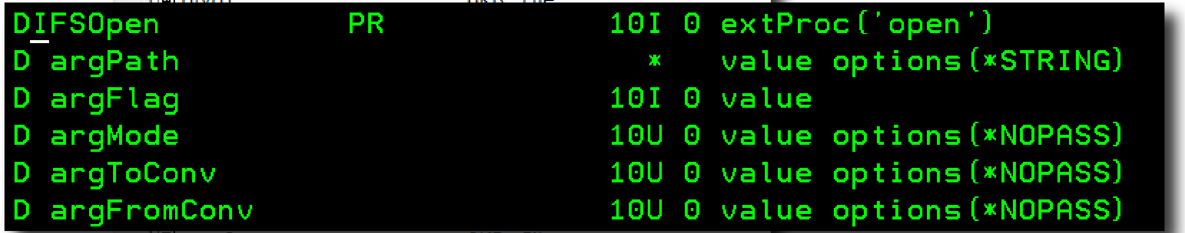
File Edit View Help

Environment: My Connections

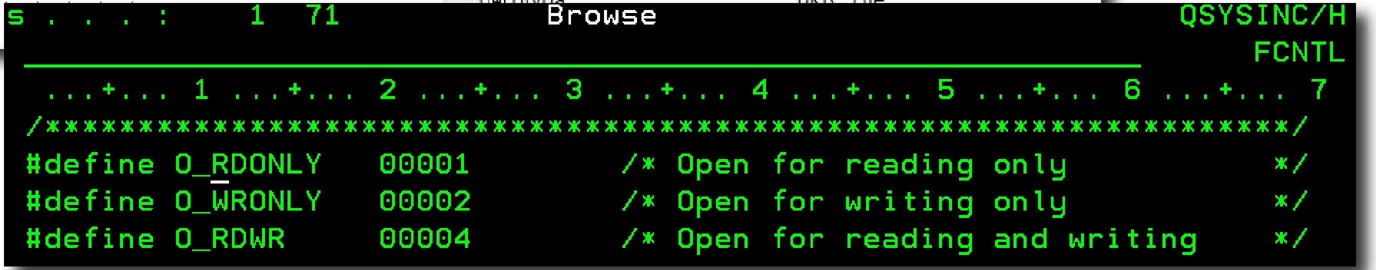
dev Include: *.*

- Configuration and Service
- Network
- Integrated Server Administration
- Security
- Users and Groups
- Databases
- File Systems
  - Integrated File System
    - Root
      - ASFTomcat
      - cgidev
      - CgidevExtHtml
      - demo
      - dev
        - pts

Name	Size	Type
console	0KB	File
null	0KB	File
ptc	0KB	File
pts		File folder
ptyp0	0KB	File
ptyp1	0KB	File
ptyp9	0KB	File
ptypa	0KB	File

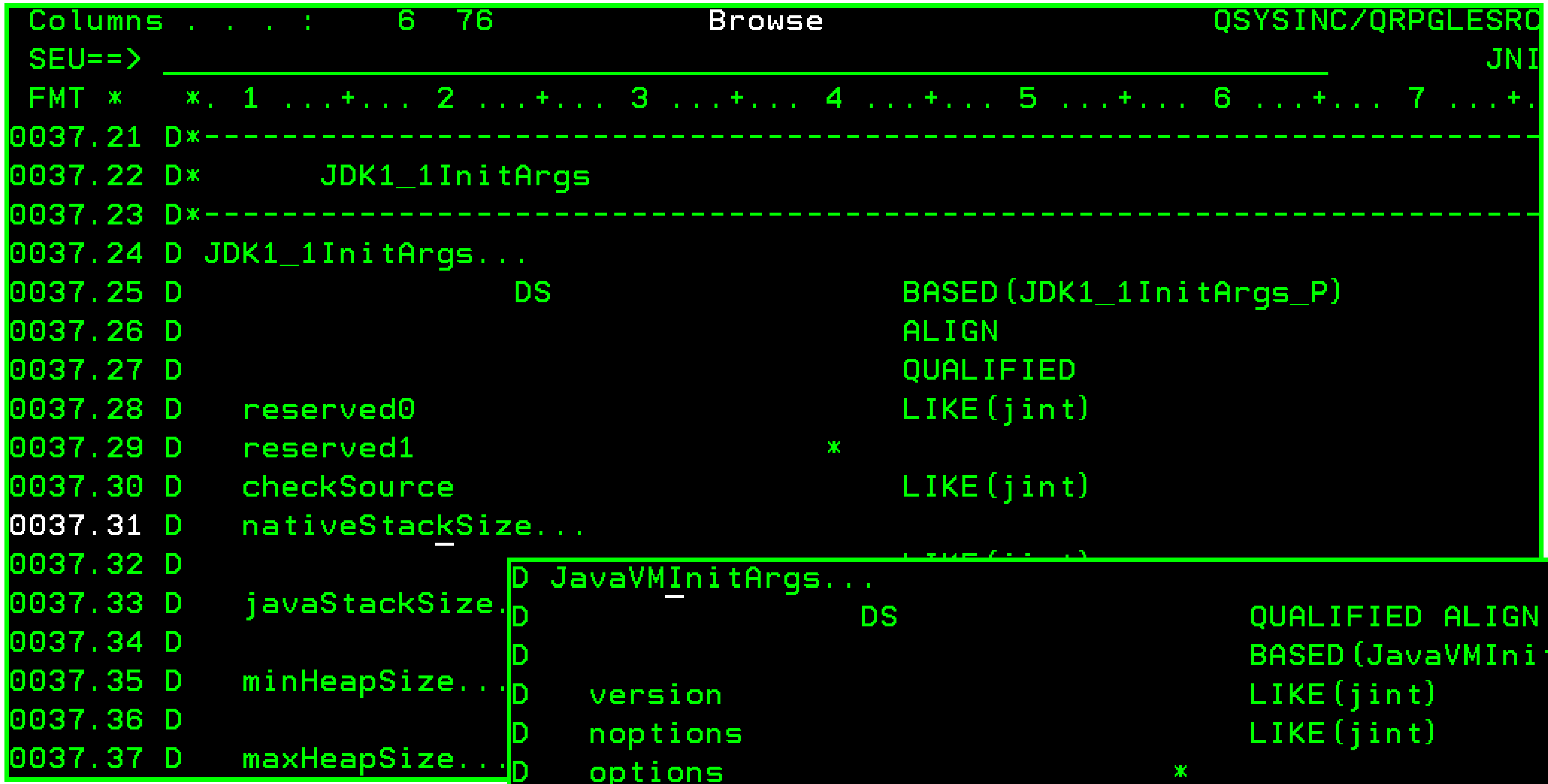


```
DIFSOpen           PR                  10I 0 extProc('open')
D argPath                               *    value options(*STRING)
D argFlag                              10I 0 value
D argMode                              10U 0 value options(*NOPASS)
D argToConv                            10U 0 value options(*NOPASS)
D argFromConv                          10U 0 value options(*NOPASS)
```

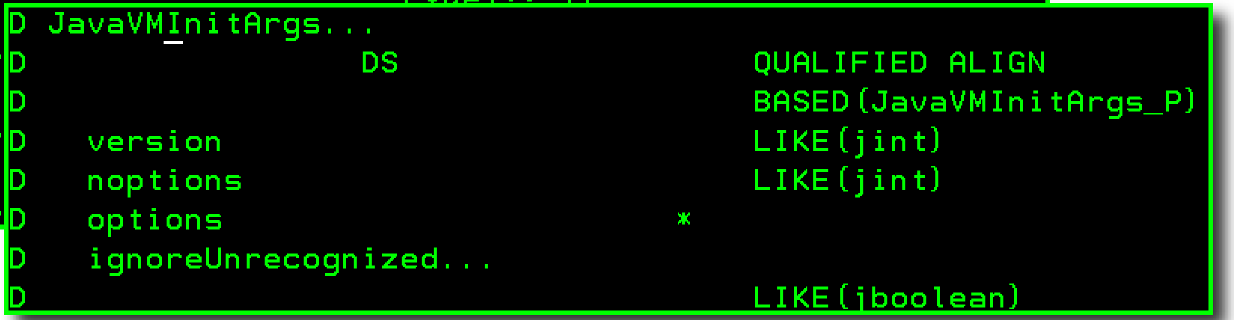


```
s . . . :     1   71              Browse                       QSYSINC/H
                                                                   FCNTL
 . . . + . . . 1 . . . + . . . 2 . . . + . . . 3 . . . + . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7
/**********************************************************************/
#define O_RDONLY    00001        /* Open for reading only           */
#define O_WRONLY    00002        /* Open for writing only           */
#define O_RDWR      00004        /* Open for reading and writing    */
```

Initialization Structures

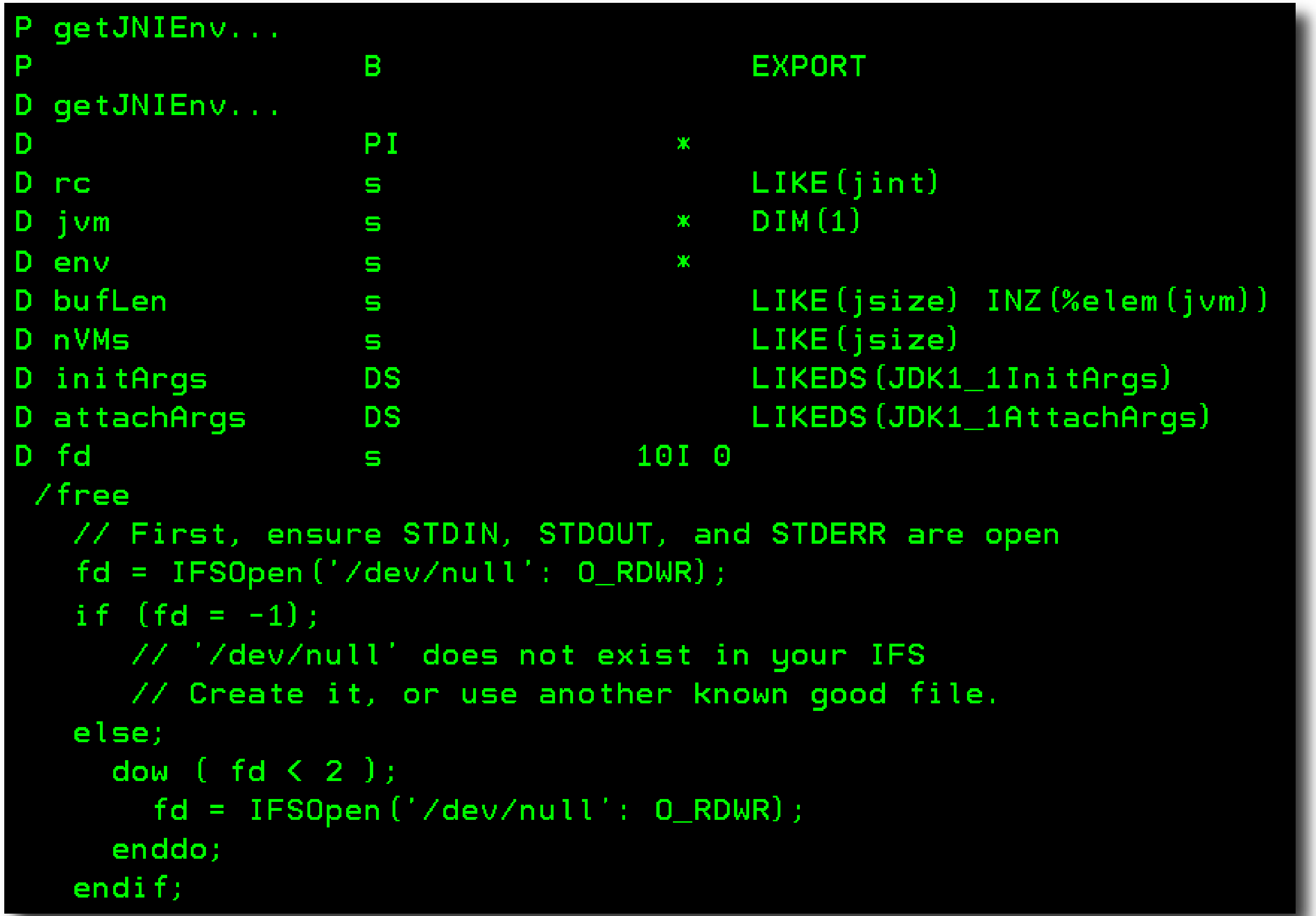


```
 Columns . . . :    6  76            Browse                    QSYSINC/QRPGLESRC
 SEU==> ______________________________________________________________       JNI
 FMT *   *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+.
 D*-------------------------------------------------------------------------
 D*      JDK1_1InitArgs
 D*-------------------------------------------------------------------------
 D JDK1_1InitArgs...
 D                 DS                    BASED(JDK1_1InitArgs_P)
 D                                       ALIGN
 D                                       QUALIFIED
 D   reserved0                           LIKE(jint)
 D   reserved1                       *
 D   checkSource                         LIKE(jint)
 D   nativeStackSize...
 D
 D   javaStackSize.
 D
 D   minHeapSize...
 D
 D   maxHeapSize...
```



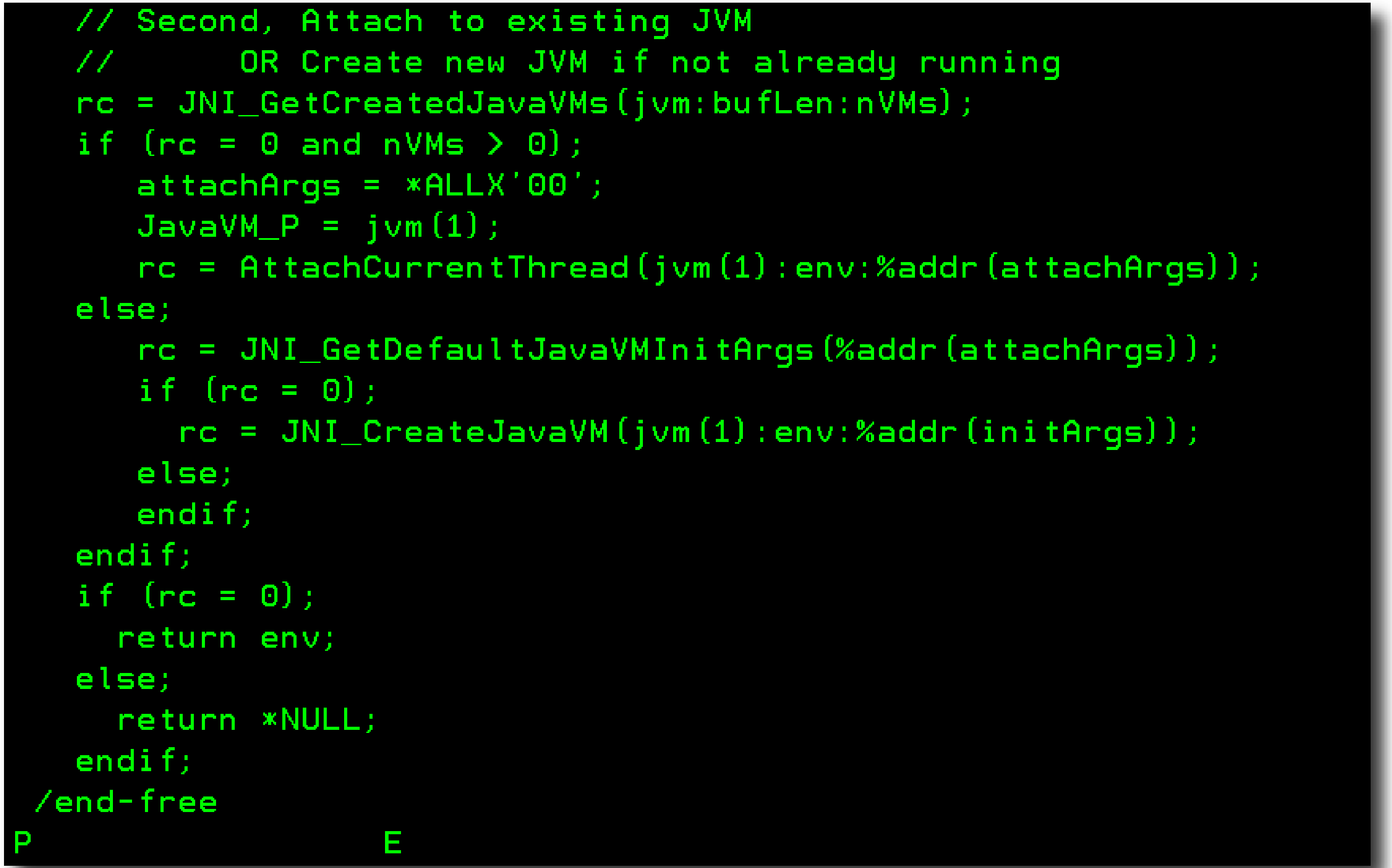
```
D JavaVMInitArgs...
D                 DS                    QUALIFIED ALIGN
D                                       BASED(JavaVMInitArgs_P)
D   version                             LIKE(jint)
D   noptions                            LIKE(jint)
D   options                         *
D   ignoreUnrecognized...
D                                       LIKE(jboolean)
```

getJNIEnv File Descriptors



```
P getJNIEnv...
P                 B                   EXPORT
D getJNIEnv...
D                 PI               *
D rc              s                   LIKE(jint)
D jvm             s                *  DIM(1)
D env             s                *
D bufLen          s                   LIKE(jsize) INZ(%elem(jvm))
D nVMs            s                   LIKE(jsize)
D initArgs        DS                  LIKEDS(JDK1_1InitArgs)
D attachArgs      DS                  LIKEDS(JDK1_1AttachArgs)
D fd              s              10I 0
 /free
   // First, ensure STDIN, STDOUT, and STDERR are open
   fd = IFSOpen('/dev/null': O_RDWR);
   if (fd = -1);
      // '/dev/null' does not exist in your IFS
      // Create it, or use another known good file.
   else;
     dow ( fd < 2 );
       fd = IFSOpen('/dev/null': O_RDWR);
     enddo;
   endif;
```

getJNIEnv Create/Attach



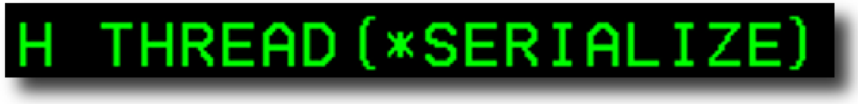
```
   // Second, Attach to existing JVM
   //      OR Create new JVM if not already running
   rc = JNI_GetCreatedJavaVMs(jvm:bufLen:nVMs);
   if (rc = 0 and nVMs > 0);
      attachArgs = *ALLX'00';
      JavaVM_P = jvm(1);
      rc = AttachCurrentThread(jvm(1):env:%addr(attachArgs));
   else;
      rc = JNI_GetDefaultJavaVMInitArgs(%addr(attachArgs));
      if (rc = 0);
        rc = JNI_CreateJavaVM(jvm(1):env:%addr(initArgs));
      else;
      endif;
   endif;
   if (rc = 0);
     return env;
   else;
     return *NULL;
   endif;
 /end-free
P                 E
```

Destroy JVM

```
P destroyJVM      B                   EXPORT
D destroyJVM      PI              N
D jvm             s                   like(JavaVM_p) dim(1)
D nVMs            s                   like(jSize)
D rc              s             10I 0
 /free
   monitor;
     rc = JNI_GetCreatedJavaVMs(jvm:1:nVMs);
     if (rc = 0 AND nVMs > 0);
       JavaVM_P = jvm(1);
       rc = DestroyJavaVM(jvm(1));
       if (rc = 0);
         return *ON;
       else;
       endif;
     else;
     endif;
   on-error;
   endmon;
   return *OFF;
 /end-free
P                 E
```

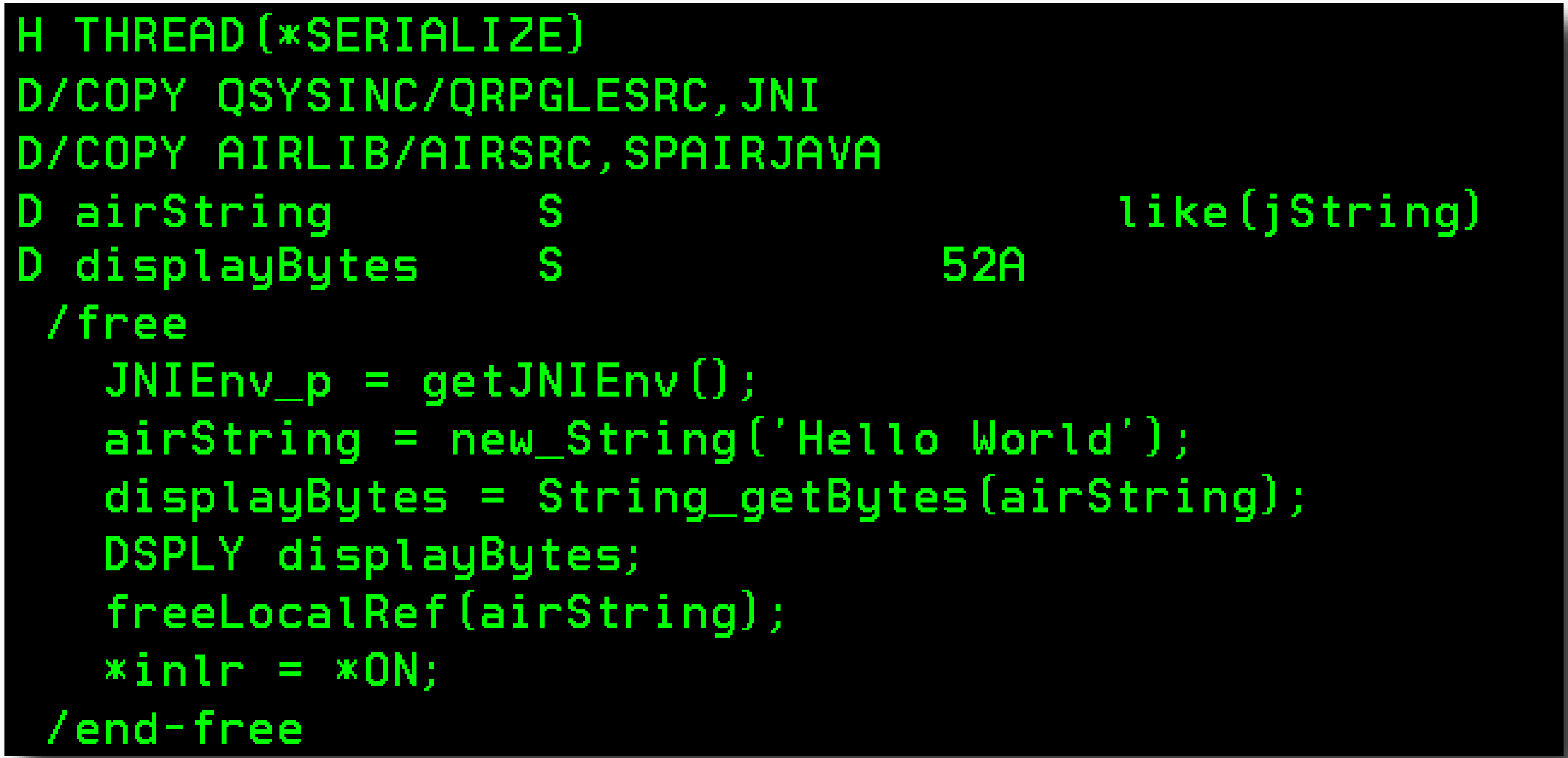
Thread Serialize

- Recommend when using Java with RPG
- Supports Multithreading
- Prevents Access to More Than One Thread at a Time

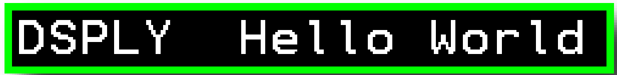


```
H THREAD(*SERIALIZE)
```

Hello World PDF



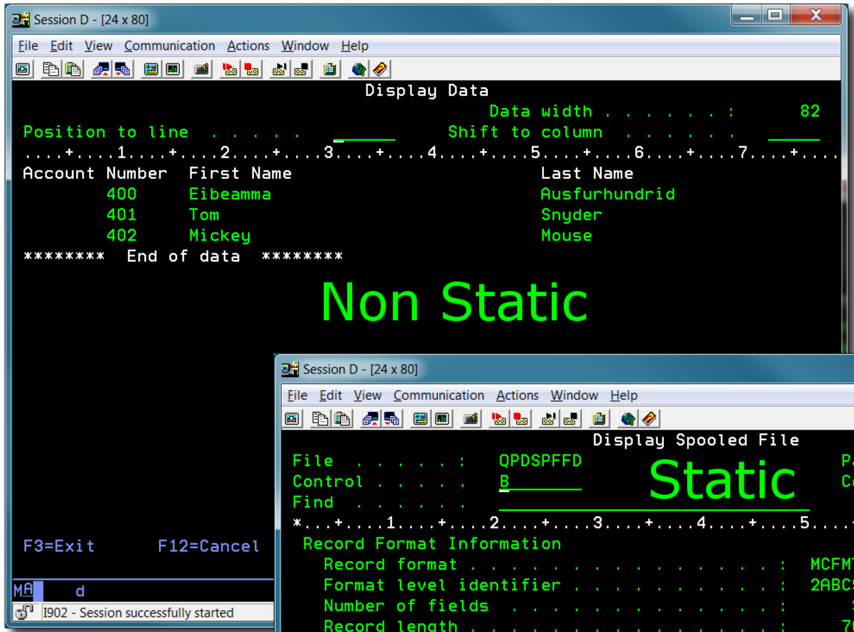
```
H THREAD(*SERIALIZE)
D/COPY QSYSINC/QRPGLESRC,JNI
D/COPY AIRLIB/AIRSRC,SPAIRJAVA
D airString       S                   like(jString)
D displayBytes    S             52A
 /free
   JNIEnv_p = getJNIEnv();
   airString = new_String('Hello World');
   displayBytes = String_getBytes(airString);
   DSPLY displayBytes;
   freeLocalRef(airString);
   *inlr = *ON;
 /end-free
```



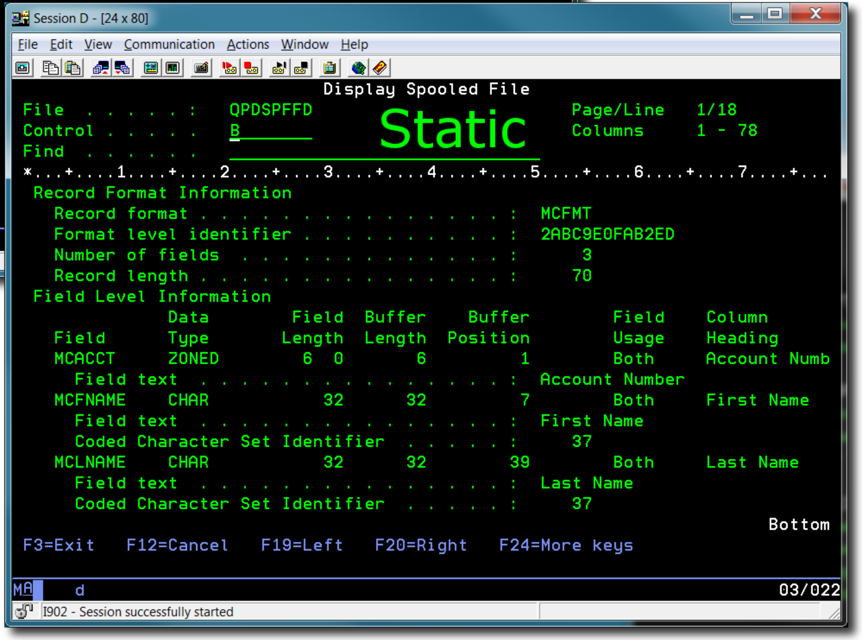
```
DSPLY  Hello World
```

Static and Non Static

Conceptual Example



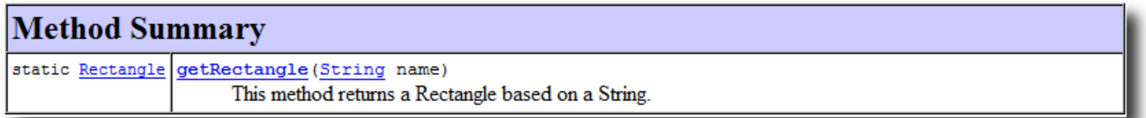
```
                                 Display Data
                                             Data width . . . . . . :      82
Position to line  . . . . .   ______     Shift to column  . . . . . .   _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
Account Number  First Name                                 Last Name
        400     Eibeamma                                   Ausfurhundrid
        401     Tom                                        Snyder
        402     Mickey                                     Mouse
********  End of data  ********

F3=Exit      F12=Cancel
MA      d
I902 - Session successfully started
```



```
                              Display Spooled File
File  . . . . . :   QPDSPFFD                       Page/Line   1/18
Control . . . . .   B_______                       Columns     1 - 78
Find  . . . . . .   ______________________________
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
 Record Format Information
   Record format . . . . . . . . . . . . . . . :  MCFMT
   Format level identifier . . . . . . . . . . :  2ABC9E0FAB2ED
   Number of fields  . . . . . . . . . . . . . :      3
   Record length . . . . . . . . . . . . . . . :     70
 Field Level Information
              Data         Field  Buffer    Buffer        Field    Column
   Field      Type        Length  Length  Position        Usage    Heading
   MCACCT     ZONED         6  0       6         1        Both     Account Numb
     Field text  . . . . . . . . . . . . . . . :  Account Number
   MCFNAME    CHAR            32      32         7        Both     First Name
     Field text  . . . . . . . . . . . . . . . :  First Name
     Coded Character Set Identifier  . . . . . :     37
   MCLNAME    CHAR            32      32        39        Both     Last Name
     Field text  . . . . . . . . . . . . . . . :  Last Name
     Coded Character Set Identifier  . . . . . :     37
                                                                         Bottom
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys

MA      d                                                               03/022
I902 - Session successfully started
```

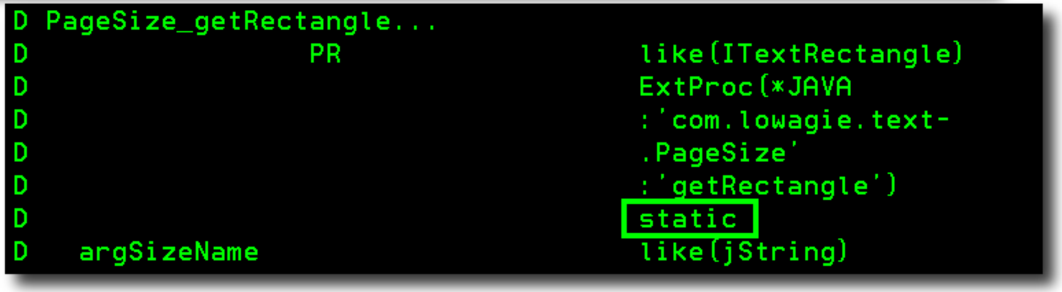
Static and Non Static

Parameter Behavior

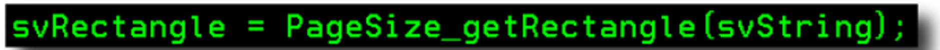


Method Summary

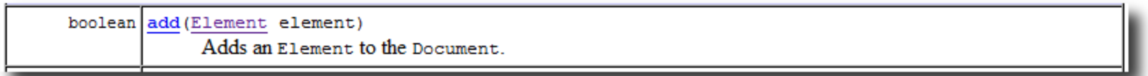
static `Rectangle`	`getRectangle(String name)` This method returns a Rectangle based on a String.



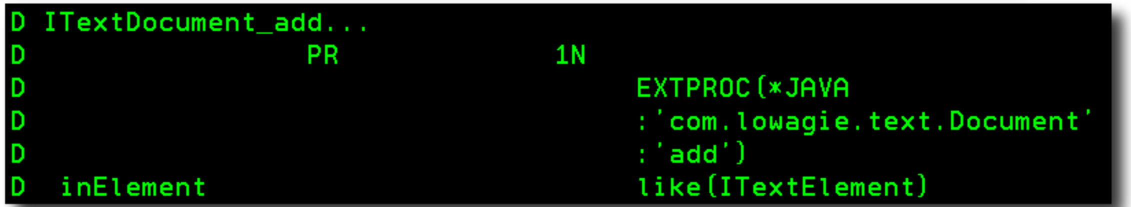
```
D PageSize_getRectangle...
D                PR                   like(ITextRectangle)
D                                     ExtProc(*JAVA
D                                     :'com.lowagie.text-
D                                     .PageSize'
D                                     :'getRectangle')
D                                     static
D  argSizeName                        like(jString)
```



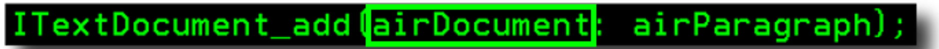
```
svRectangle = PageSize_getRectangle(svString);
```



boolean	`add(Element element)` Adds an `Element` to the `Document`.



```
D ITextDocument_add...
D                PR           1N
D                                     EXTPROC(*JAVA
D                                     :'com.lowagie.text.Document'
D                                     :'add')
D  inElement                          like(ITextElement)
```

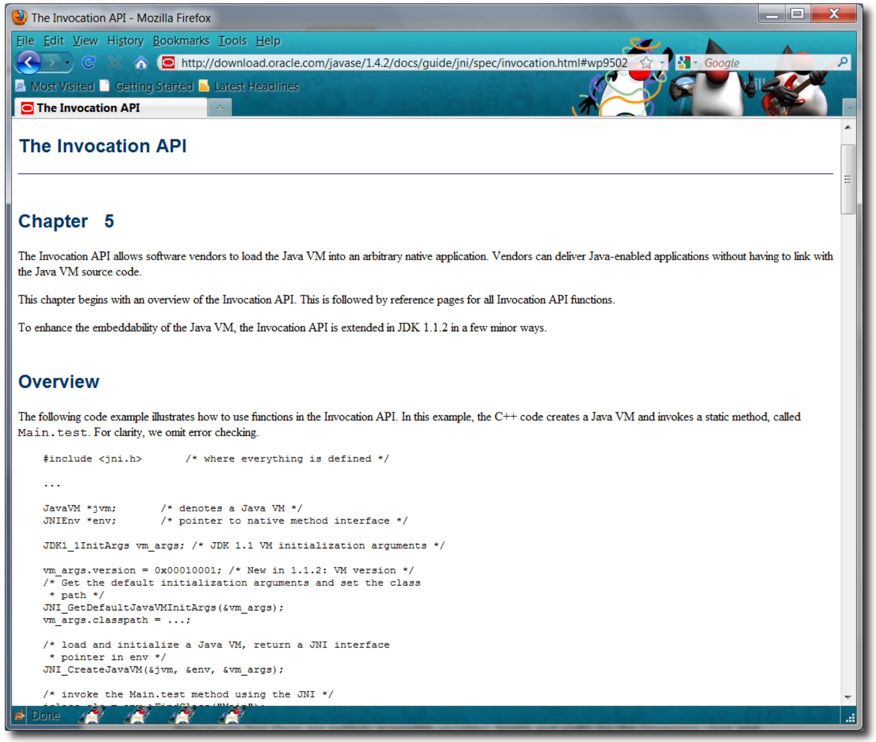


```
ITextDocument_add(airDocument: airParagraph);
```

Java Native Interface (JNI)

- ➢Sun/Oracle Java Native Interface (JNI) Specifications

- ➢ILE RPG Programmer’s Guide SC09-2507-06


The Invocation API - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://download.oracle.com/javase/1.4.2/docs/guide/jni/spec/invocation.html#wp9502
Google
Most Visited Getting Started Latest Headlines
The Invocation API
Done

The Invocation API

Chapter 5

The Invocation API allows software vendors to load the Java VM into an arbitrary native application. Vendors can deliver Java-enabled applications without having to link with the Java VM source code.

This chapter begins with an overview of the Invocation API. This is followed by reference pages for all Invocation API functions.

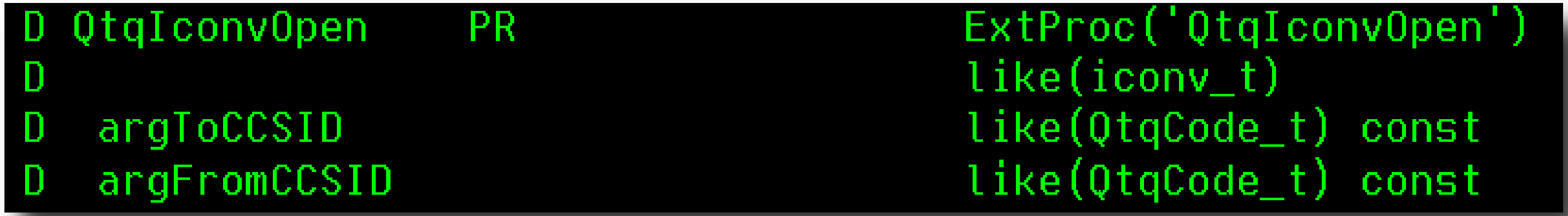
To enhance the embeddability of the Java VM, the Invocation API is extended in JDK 1.1.2 in a few minor ways.

Overview

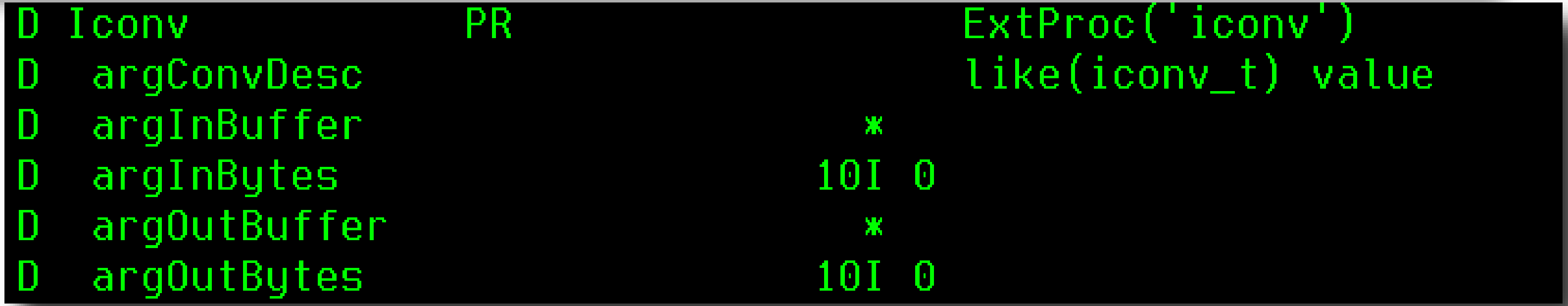
The following code example illustrates how to use functions in the Invocation API. In this example, the C++ code creates a Java VM and invokes a static method, called `Main.test`. For clarity, we omit error checking.

```
    #include <jni.h>       /* where everything is defined */

    ...

    JavaVM *jvm;       /* denotes a Java VM */
    JNIEnv *env;       /* pointer to native method interface */

    JDK1_1InitArgs vm_args; /* JDK 1.1 VM initialization arguments */

    vm_args.version = 0x00010001; /* New in 1.1.2: VM version */
    /* Get the default initialization arguments and set the class
     * path */
    JNI_GetDefaultJavaVMInitArgs(&vm_args);
    vm_args.classpath = ...;

    /* load and initialize a Java VM, return a JNI interface
     * pointer in env */
    JNI_CreateJavaVM(&jvm, &env, &vm_args);

    /* invoke the Main.test method using the JNI */
```

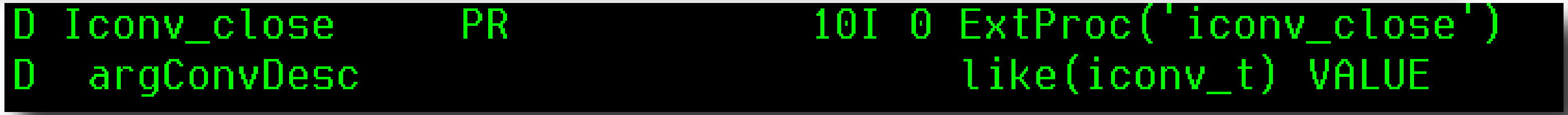
Conversion Descriptor - iconv



```
D QtqIconvOpen    PR                   ExtProc('QtqIconvOpen')
D                                      like(iconv_t)
D  argToCCSID                          like(QtqCode_t) const
D  argFromCCSID                        like(QtqCode_t) const
```

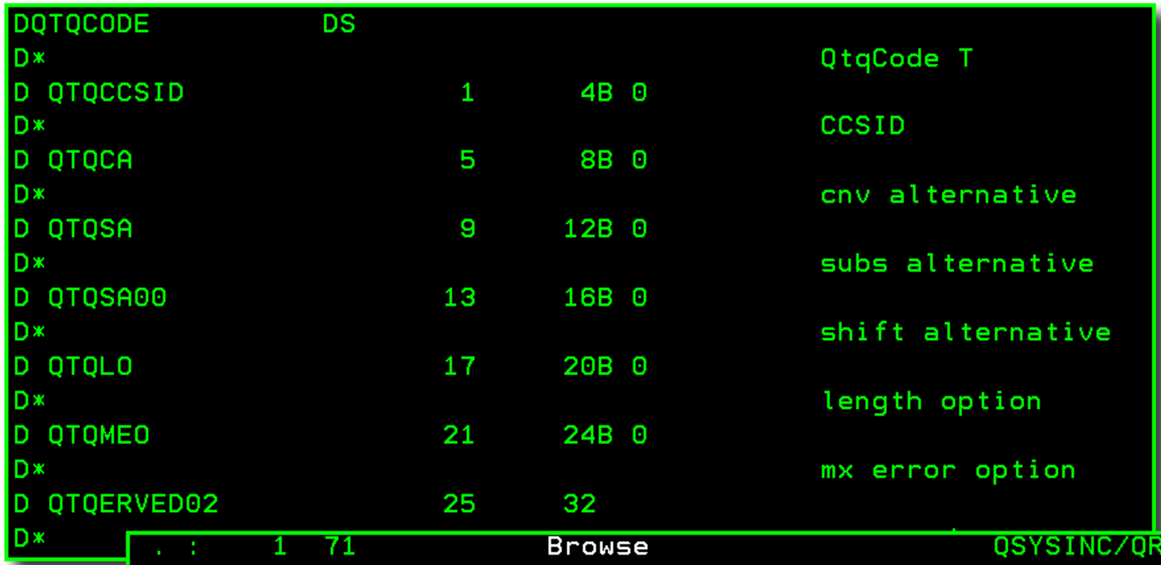


```
D Iconv           PR                   ExtProc('iconv')
D  argConvDesc                         like(iconv_t) value
D  argInBuffer                    *
D  argInBytes                   10I 0
D  argOutBuffer                   *
D  argOutBytes                  10I 0
```

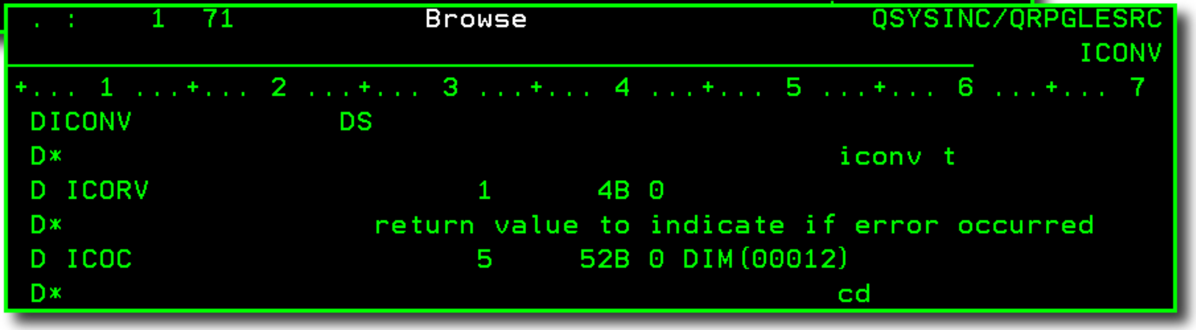


```
D Iconv_close     PR            10I 0 ExtProc('iconv_close')
D  argConvDesc                        like(iconv_t) VALUE
```

iconv Data Structures

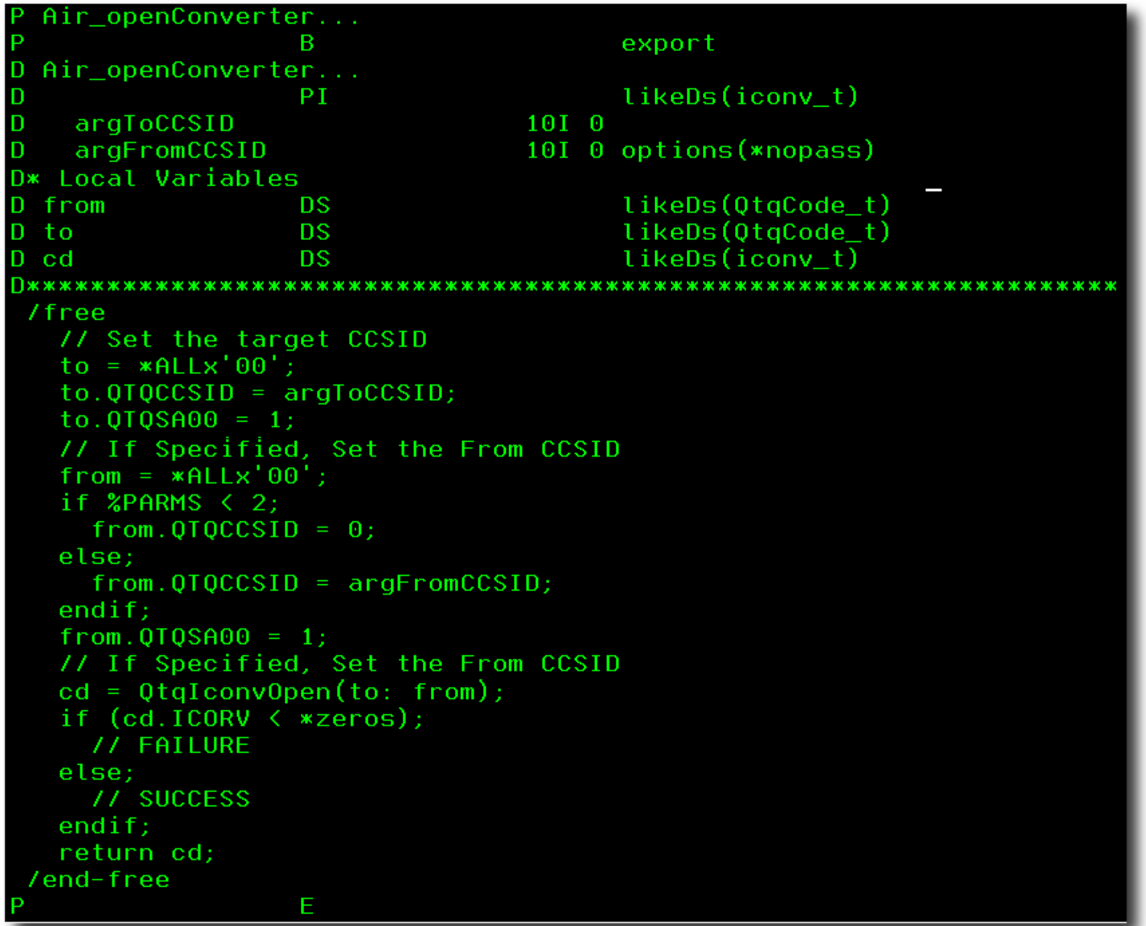


```
DQTQCODE          DS
D*                                                QtqCode T
D QTQCCSID                1      4B 0
D*                                                CCSID
D QTQCA                   5      8B 0
D*                                                cnv alternative
D QTQSA                   9     12B 0
D*                                                subs alternative
D QTQSA00                13     16B 0
D*                                                shift alternative
D QTQLO                  17     20B 0
D*                                                length option
D QTQMEO                 21     24B 0
D*                                                mx error option
D QTQERVED02             25     32
D*
```



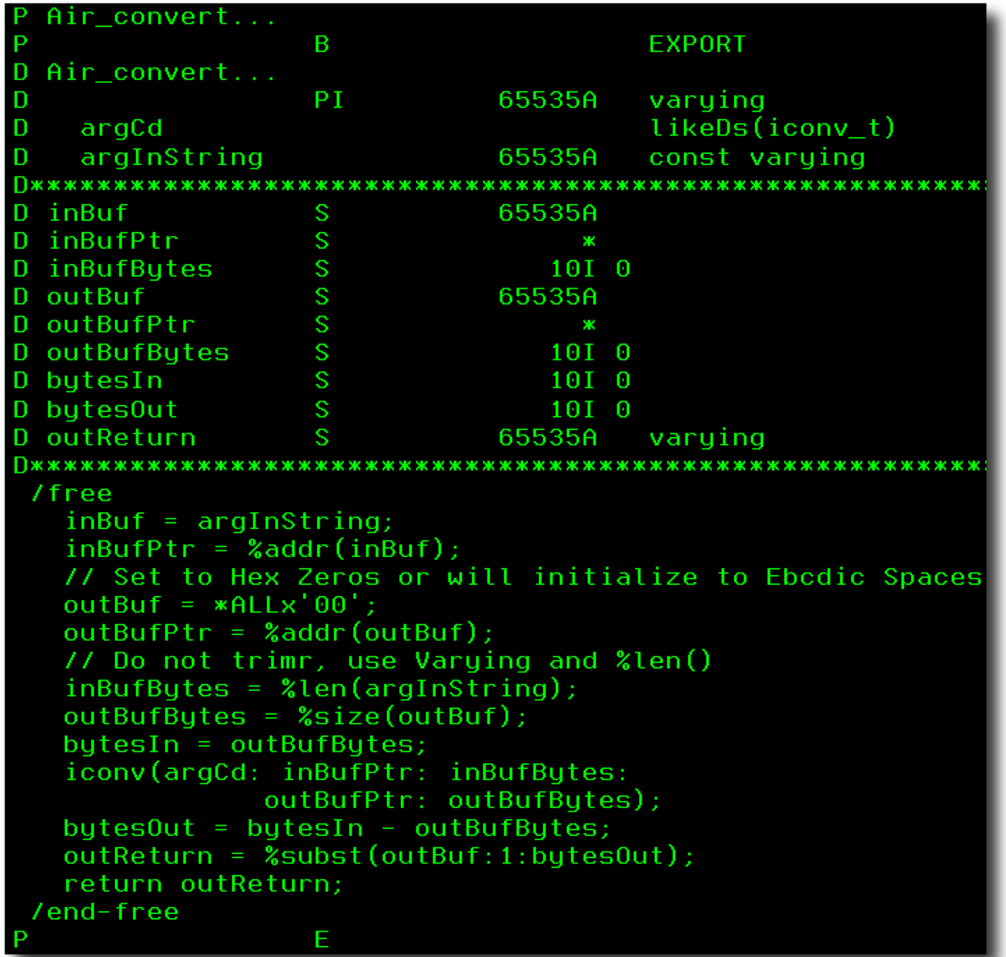
```
 . :    1  71              Browse                  QSYSINC/QRPGLESRC
                                                               ICONV
+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
 DICONV            DS
 D*                                               iconv t
 D ICORV                   1      4B 0
 D*                    return value to indicate if error occurred
 D ICOC                    5     52B 0 DIM(00012)
 D*                                               cd
```

Air_openConverter Procedure



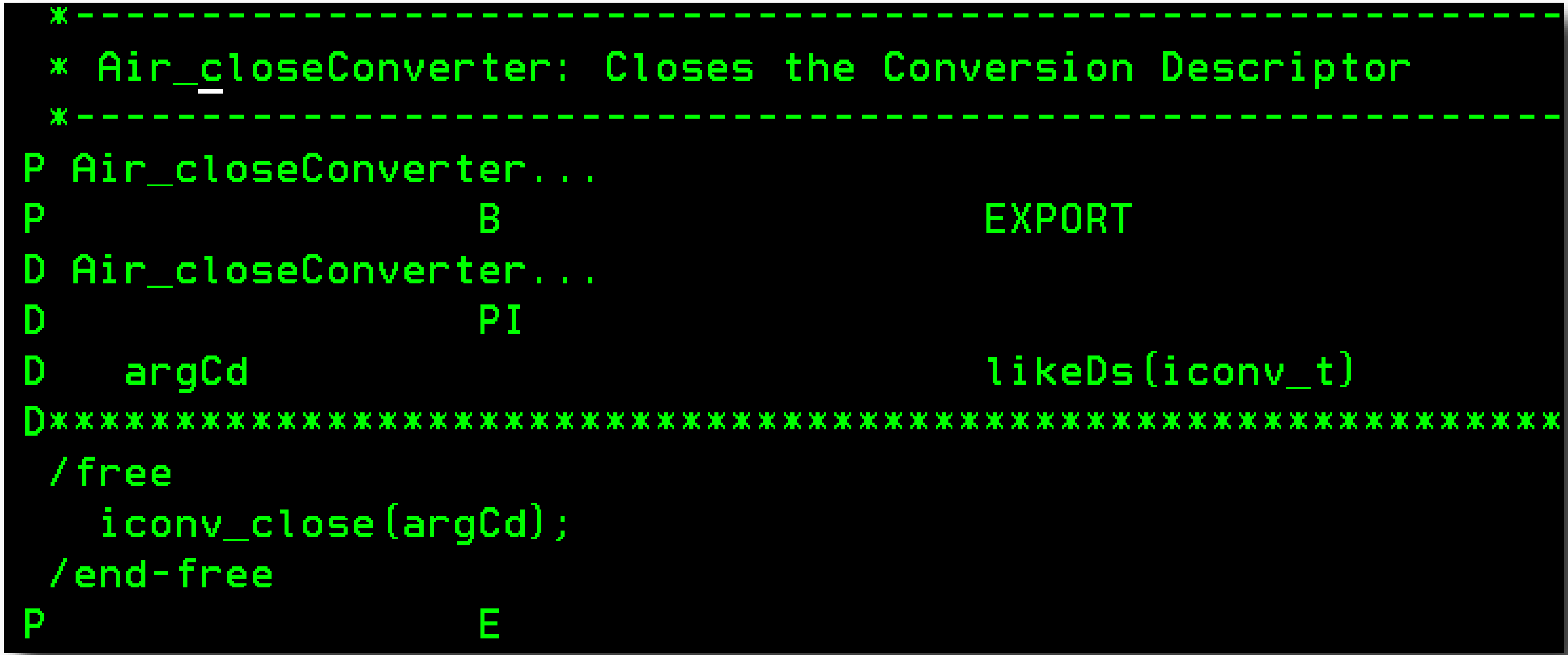
```
P Air_openConverter...
P                 B                   export
D Air_openConverter...
D                 PI                  likeDs(iconv_t)
D   argToCCSID                  10I 0
D   argFromCCSID                10I 0 options(*nopass)
D* Local Variables
D from            DS                  likeDs(QtqCode_t)
D to              DS                  likeDs(QtqCode_t)
D cd              DS                  likeDs(iconv_t)
D*********************************************************************
 /free
   // Set the target CCSID
   to = *ALLx'00';
   to.QTQCCSID = argToCCSID;
   to.QTQSA00 = 1;
   // If Specified, Set the From CCSID
   from = *ALLx'00';
   if %PARMS < 2;
     from.QTQCCSID = 0;
   else;
     from.QTQCCSID = argFromCCSID;
   endif;
   from.QTQSA00 = 1;
   // If Specified, Set the From CCSID
   cd = QtqIconvOpen(to: from);
   if (cd.ICORV < *zeros);
     // FAILURE
   else;
     // SUCCESS
   endif;
   return cd;
 /end-free
P                 E
```

Air_convert Procedure



```
P Air_convert...
P                 B                   EXPORT
D Air_convert...
D                 PI          65535A   varying
D   argCd                              likeDs(iconv_t)
D   argInString               65535A   const varying
D*********************************************************
D inBuf           S           65535A
D inBufPtr        S                *
D inBufBytes      S              10I 0
D outBuf          S           65535A
D outBufPtr       S                *
D outBufBytes     S              10I 0
D bytesIn         S              10I 0
D bytesOut        S              10I 0
D outReturn       S           65535A   varying
D*********************************************************
 /free
   inBuf = argInString;
   inBufPtr = %addr(inBuf);
   // Set to Hex Zeros or will initialize to Ebcdic Spaces
   outBuf = *ALLx'00';
   outBufPtr = %addr(outBuf);
   // Do not trimr, use Varying and %len()
   inBufBytes = %len(argInString);
   outBufBytes = %size(outBuf);
   bytesIn = outBufBytes;
   iconv(argCd: inBufPtr: inBufBytes:
               outBufPtr: outBufBytes);
   bytesOut = bytesIn - outBufBytes;
   outReturn = %subst(outBuf:1:bytesOut);
   return outReturn;
 /end-free
P                 E
```

Air_closeConverter Procedure



```
 *----------------------------------------------------------
 * Air_closeConverter: Closes the Conversion Descriptor
 *----------------------------------------------------------
P Air_closeConverter...
P                 B                   EXPORT
D Air_closeConverter...
D                 PI
D   argCd                             likeDs(iconv_t)
D***********************************************************
 /free
   iconv_close(argCd);
 /end-free
P                 E
```

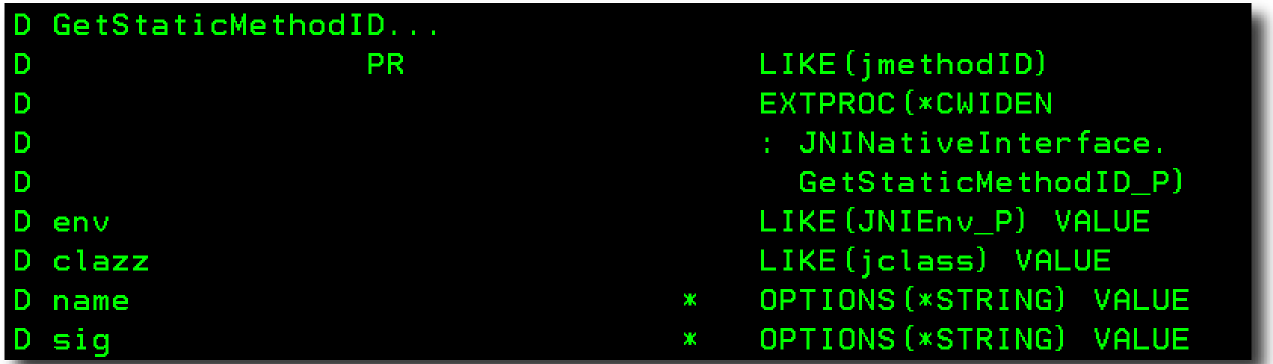
QSYSINC/QRPGLESRC, JNI

FindClass Prototype

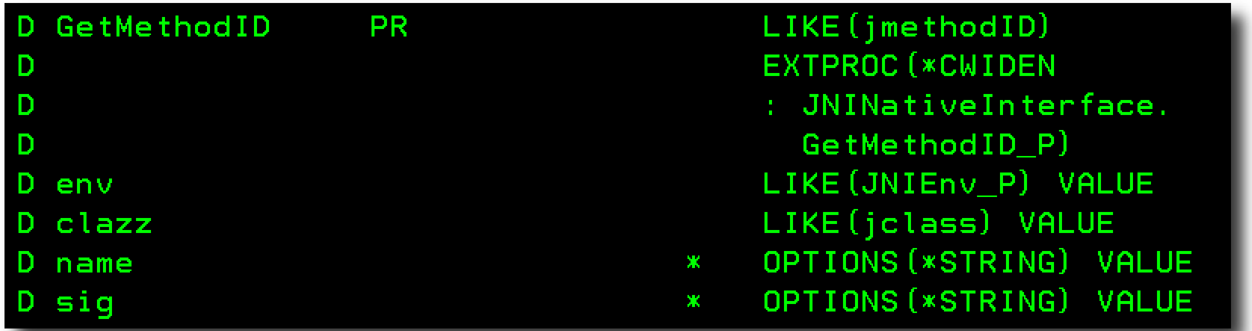
```
D*------------------------------------------------------------
D*      jclass (*FindClass)
D*        (JNIEnv *env, const char *name);
D*------------------------------------------------------------
D FindClass        PR                   LIKE(jclass)
D                                       EXTPROC(*CWIDEN
D                                       : JNINativeInterface.
D                                         FindClass_P)
D env                                   LIKE(JNIEnv_P) VALUE
D name                              *   OPTIONS(*STRING) VALUE
```

QSYSINC/QRPGLESRC, JNI

GetStaticMethodID, GetMethodID

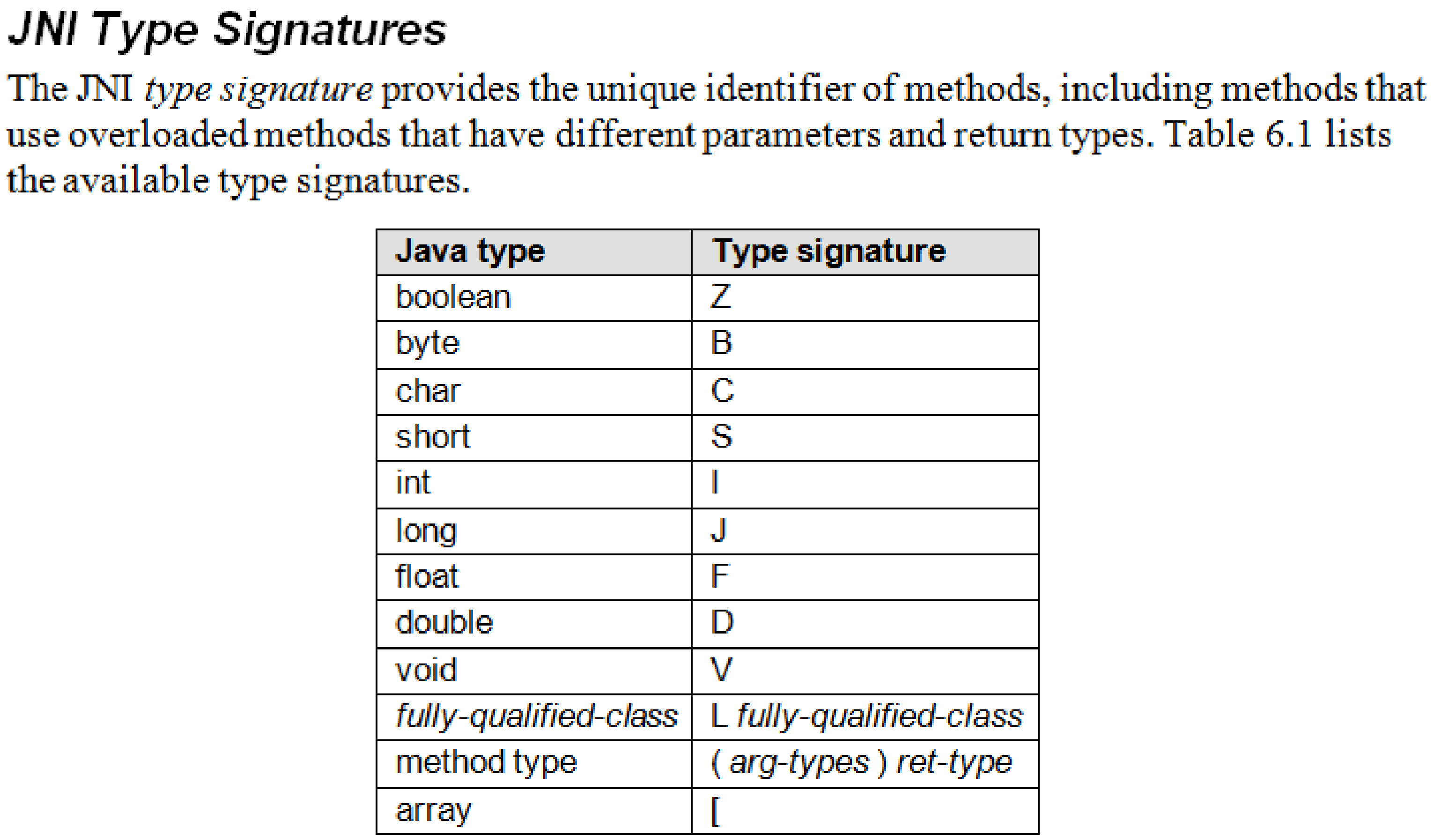


```
D GetStaticMethodID...
D                 PR                  LIKE(jmethodID)
D                                     EXTPROC(*CWIDEN
D                                     : JNINativeInterface.
D                                       GetStaticMethodID_P)
D env                                 LIKE(JNIEnv_P) VALUE
D clazz                               LIKE(jclass) VALUE
D name                            *   OPTIONS(*STRING) VALUE
D sig                             *   OPTIONS(*STRING) VALUE
```



```
D GetMethodID     PR                  LIKE(jmethodID)
D                                     EXTPROC(*CWIDEN
D                                     : JNINativeInterface.
D                                       GetMethodID_P)
D env                                 LIKE(JNIEnv_P) VALUE
D clazz                               LIKE(jclass) VALUE
D name                            *   OPTIONS(*STRING) VALUE
D sig                             *   OPTIONS(*STRING) VALUE
```

JNI Type Signatures

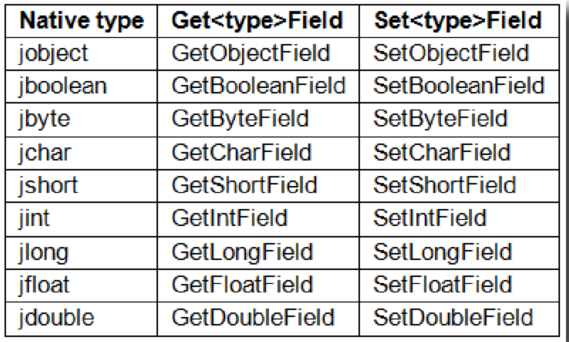


JNI Type Signatures

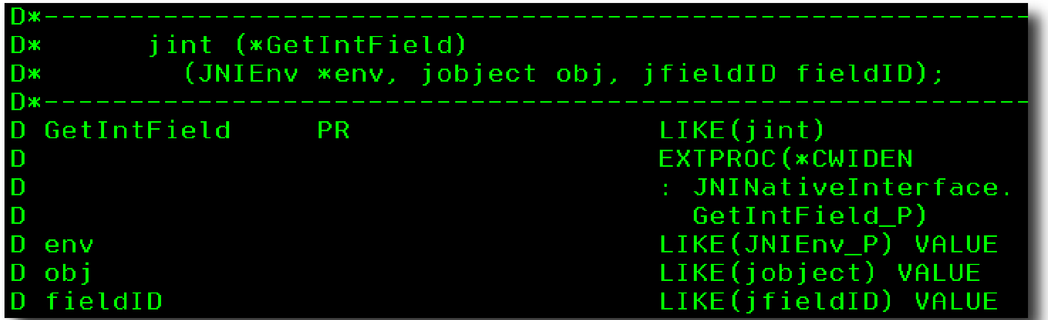
The JNI *type signature* provides the unique identifier of methods, including methods that use overloaded methods that have different parameters and return types. Table 6.1 lists the available type signatures.

Java type	Type signature
boolean	Z
byte	B
char	C
short	S
int	I
long	J
float	F
double	D
void	V
fully-qualified-class	L *fully-qualified-class*
method type	(*arg-types*) *ret-type*
array	[

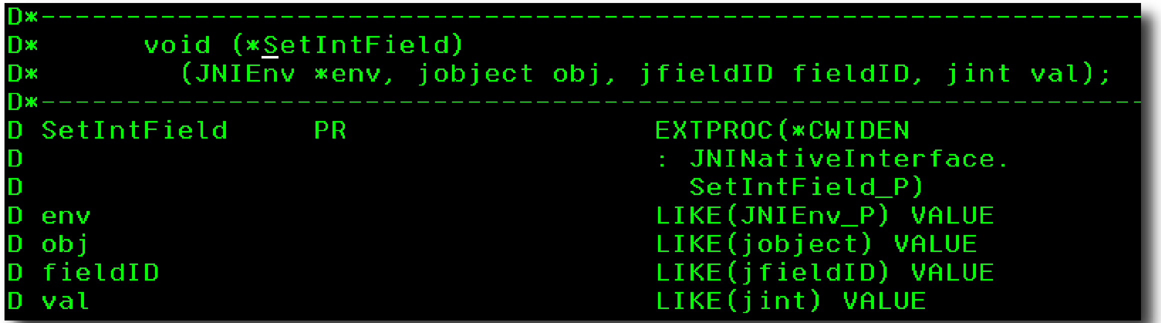
Get/Set<type>Field



Native type	Get<type>Field	Set<type>Field
jobject	GetObjectField	SetObjectField
jboolean	GetBooleanField	SetBooleanField
jbyte	GetByteField	SetByteField
jchar	GetCharField	SetCharField
jshort	GetShortField	SetShortField
jint	GetIntField	SetIntField
jlong	GetLongField	SetLongField
jfloat	GetFloatField	SetFloatField
jdouble	GetDoubleField	SetDoubleField



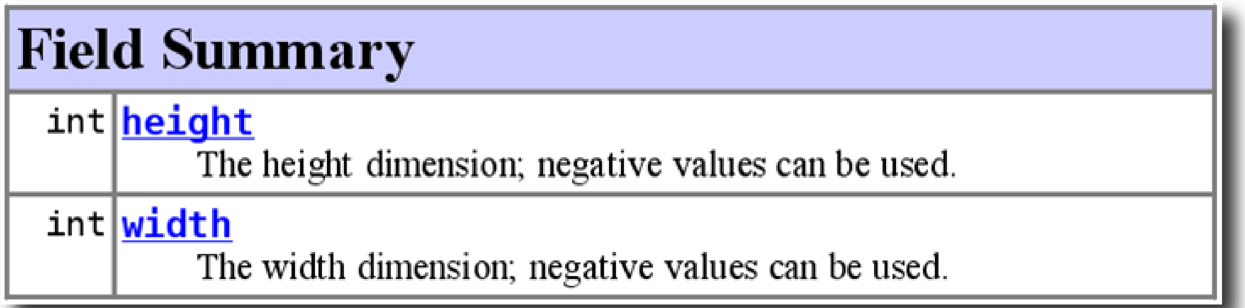
```
D*---------------------------------------------------------------
D*       jint (*GetIntField)
D*         (JNIEnv *env, jobject obj, jfieldID fieldID);
D*---------------------------------------------------------------
D GetIntField     PR                        LIKE(jint)
D                                           EXTPROC(*CWIDEN
D                                           : JNINativeInterface.
D                                             GetIntField_P)
D env                                       LIKE(JNIEnv_P) VALUE
D obj                                       LIKE(jobject) VALUE
D fieldID                                   LIKE(jfieldID) VALUE
```



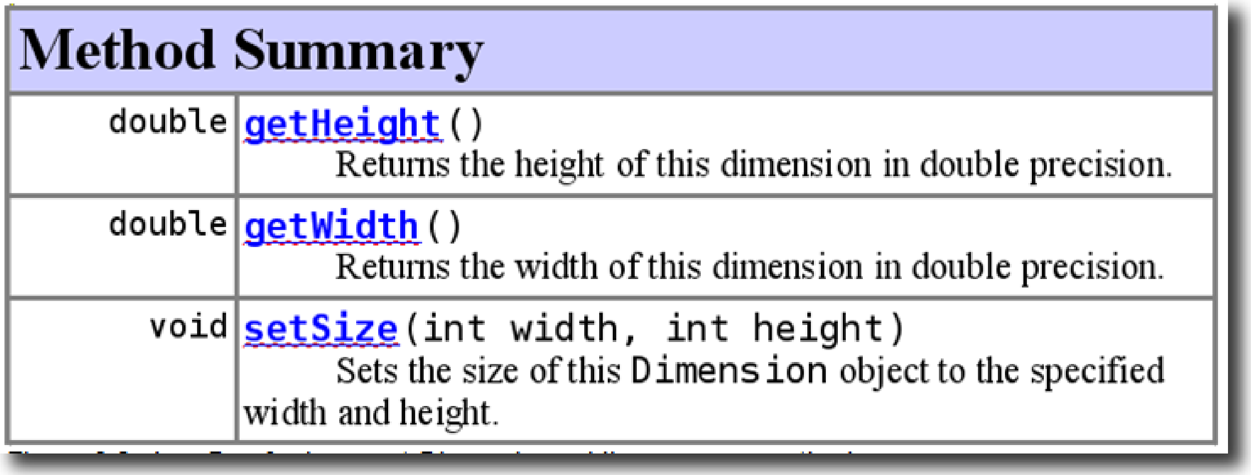
```
D*-------------------------------------------------------------------
D*       void (*SetIntField)
D*         (JNIEnv *env, jobject obj, jfieldID fieldID, jint val);
D*-------------------------------------------------------------------
D SetIntField     PR                        EXTPROC(*CWIDEN
D                                           : JNINativeInterface.
D                                             SetIntField_P)
D env                                       LIKE(JNIEnv_P) VALUE
D obj                                       LIKE(jobject) VALUE
D fieldID                                   LIKE(jfieldID) VALUE
D val                                       LIKE(jint) VALUE
```

Dimension Class

Public Fields and Methods



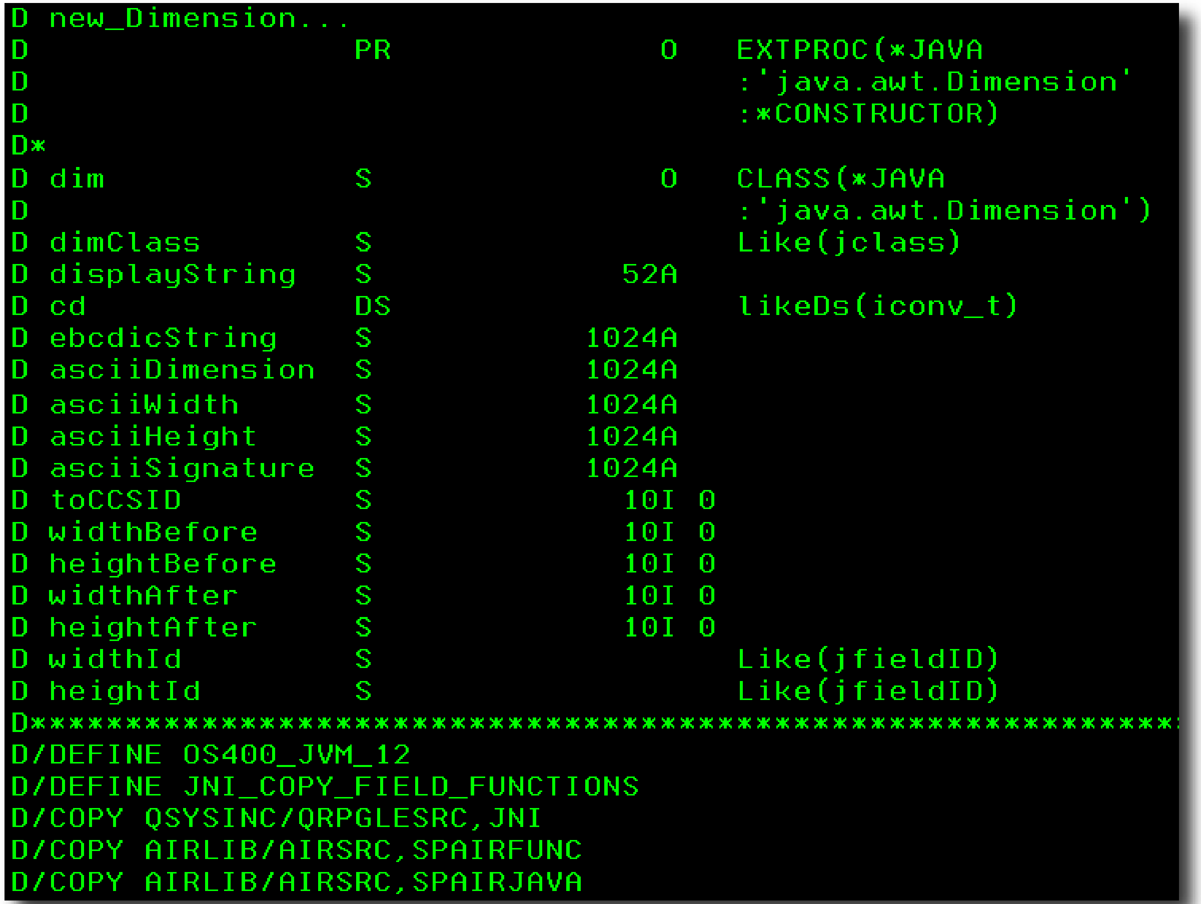
Field Summary	
int	height The height dimension; negative values can be used.
int	width The width dimension; negative values can be used.



Method Summary	
double	getHeight() Returns the height of this dimension in double precision.
double	getWidth() Returns the width of this dimension in double precision.
void	setSize(int width, int height) Sets the size of this Dimension object to the specified width and height.

JNI Code Example

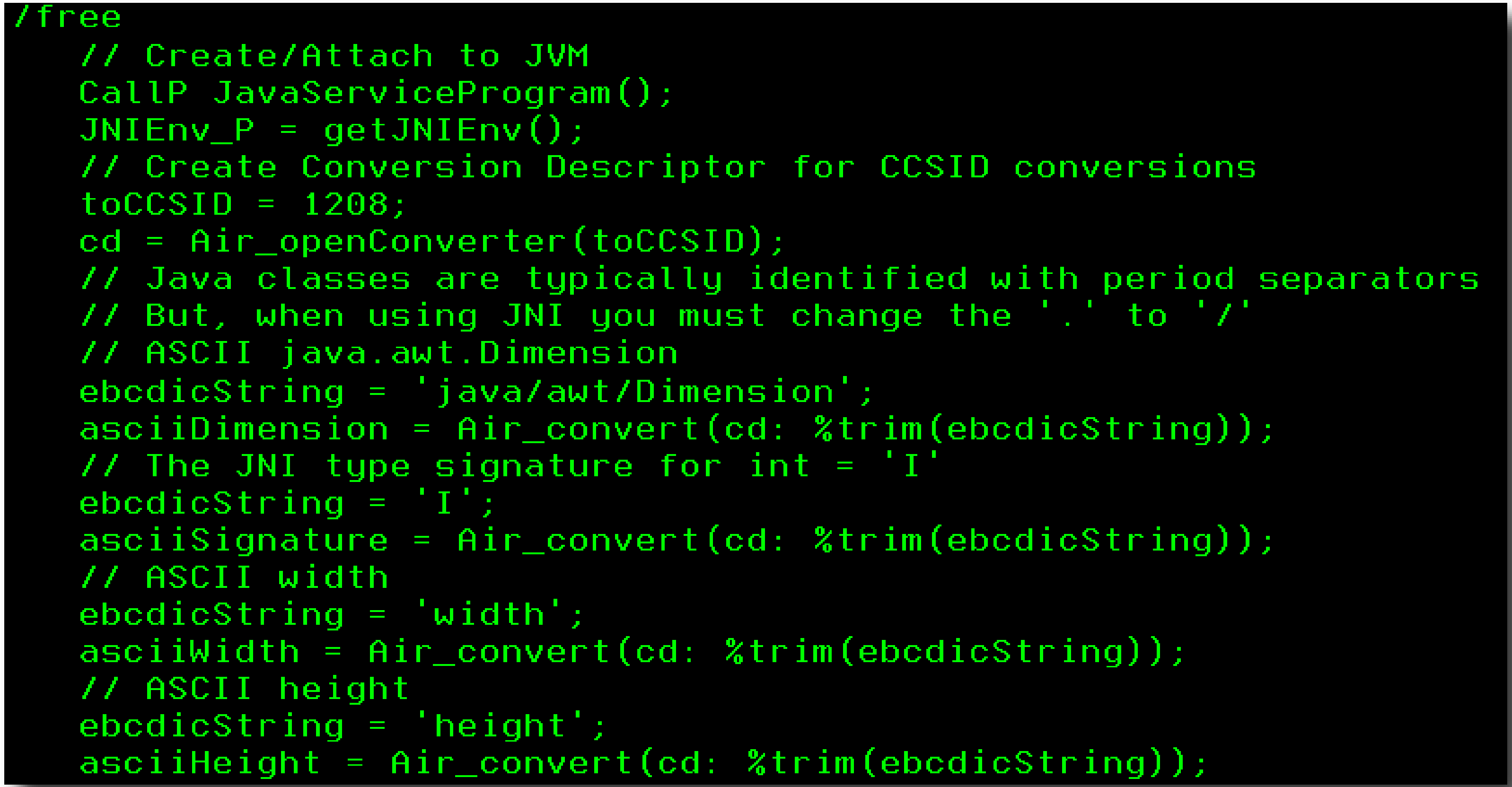
Prototypes, Variables and COPYs



```
D new_Dimension...
D                 PR              0   EXTPROC(*JAVA
D                                     :'java.awt.Dimension'
D                                     :*CONSTRUCTOR)
D*
D dim             S               0   CLASS(*JAVA
D                                     :'java.awt.Dimension')
D dimClass        S                   Like(jclass)
D displayString   S             52A
D cd              DS                  likeDs(iconv_t)
D ebcdicString    S           1024A
D asciiDimension  S           1024A
D asciiWidth      S           1024A
D asciiHeight     S           1024A
D asciiSignature  S           1024A
D toCCSID         S             10I 0
D widthBefore     S             10I 0
D heightBefore    S             10I 0
D widthAfter      S             10I 0
D heightAfter     S             10I 0
D widthId         S                   Like(jfieldID)
D heightId        S                   Like(jfieldID)
D************************************************************
D/DEFINE OS400_JVM_12
D/DEFINE JNI_COPY_FIELD_FUNCTIONS
D/COPY QSYSINC/QRPGLESRC,JNI
D/COPY AIRLIB/AIRSRC,SPAIRFUNC
D/COPY AIRLIB/AIRSRC,SPAIRJAVA
```

JNI Code Example

Converting from EBCDIC to ASCII



```
/free
   // Create/Attach to JVM
   CallP JavaServiceProgram();
   JNIEnv_P = getJNIEnv();
   // Create Conversion Descriptor for CCSID conversions
   toCCSID = 1208;
   cd = Air_openConverter(toCCSID);
   // Java classes are typically identified with period separators
   // But, when using JNI you must change the '.' to '/'
   // ASCII java.awt.Dimension
   ebcdicString = 'java/awt/Dimension';
   asciiDimension = Air_convert(cd: %trim(ebcdicString));
   // The JNI type signature for int = 'I'
   ebcdicString = 'I';
   asciiSignature = Air_convert(cd: %trim(ebcdicString));
   // ASCII width
   ebcdicString = 'width';
   asciiWidth = Air_convert(cd: %trim(ebcdicString));
   // ASCII height
   ebcdicString = 'height';
   asciiHeight = Air_convert(cd: %trim(ebcdicString));
```

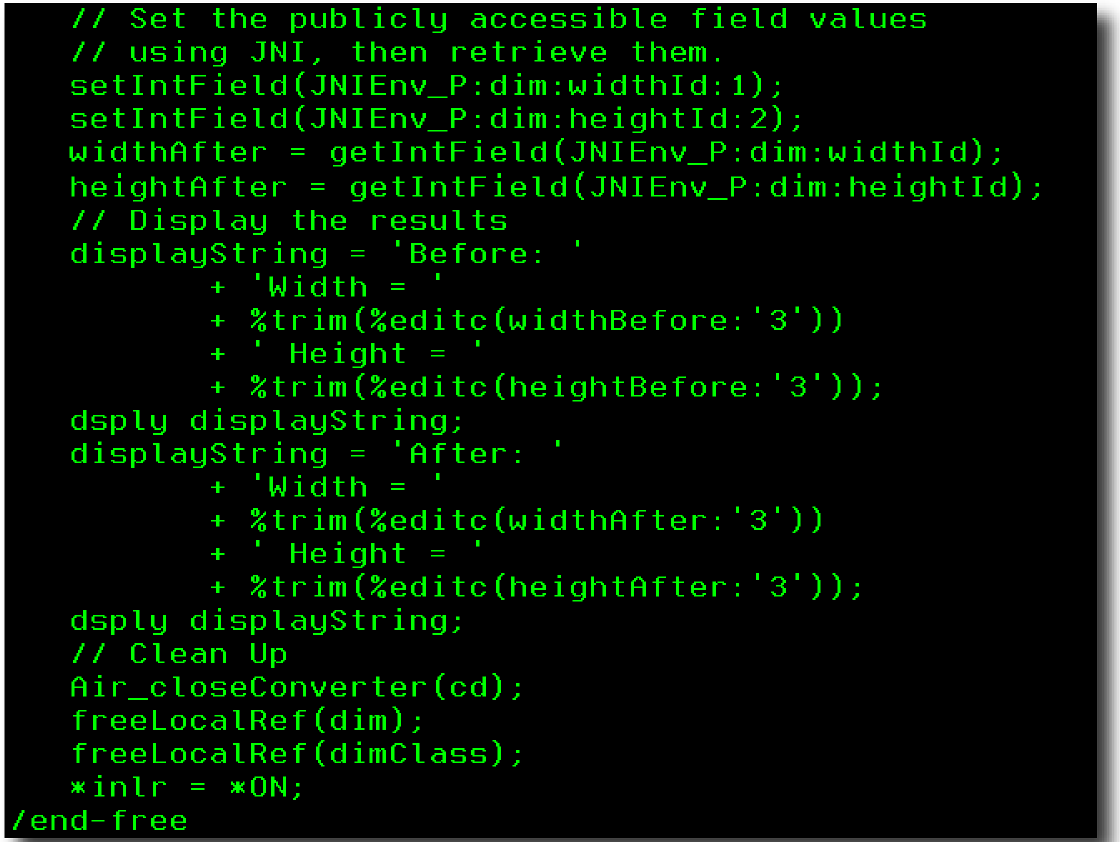
JNI Code Example

Finding/Retrieving the Class and Fields

```
// Get an instance of the Dimension Class
dim = new_Dimension();
// Get the Class reference using JNI
dimClass = FindClass(JNIEnv_P:%trim(asciiDimension));
if (dimClass = *null);
  displayString = 'Dimension FindClass Error';
  dsply displayString;
else;
endif;
// Get the Field references within the Class
widthId = GetFieldID(JNIEnv_P:dimClass:
                     %trim(asciiWidth):
                     %trim(asciiSignature));
heightId = GetFieldID(JNIEnv_P:dimClass:
                     %trim(asciiHeight):
                     %trim(asciiSignature));
// Retrieve the publicly accessible field values
// of the Dimension instance on Initialization
widthBefore = getIntField(JNIEnv_P:dim:widthId);
heightBefore = getIntField(JNIEnv_P:dim:heightId);
```

JNI Code Example

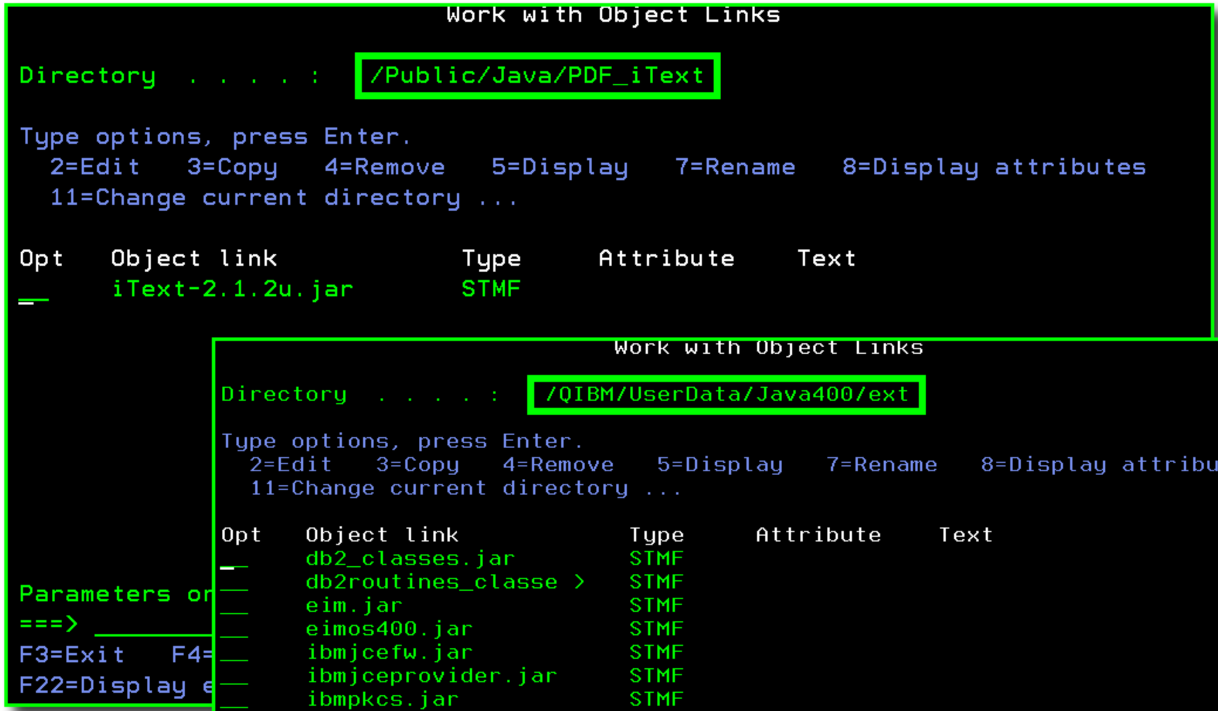
Setting and Displaying Public Fields



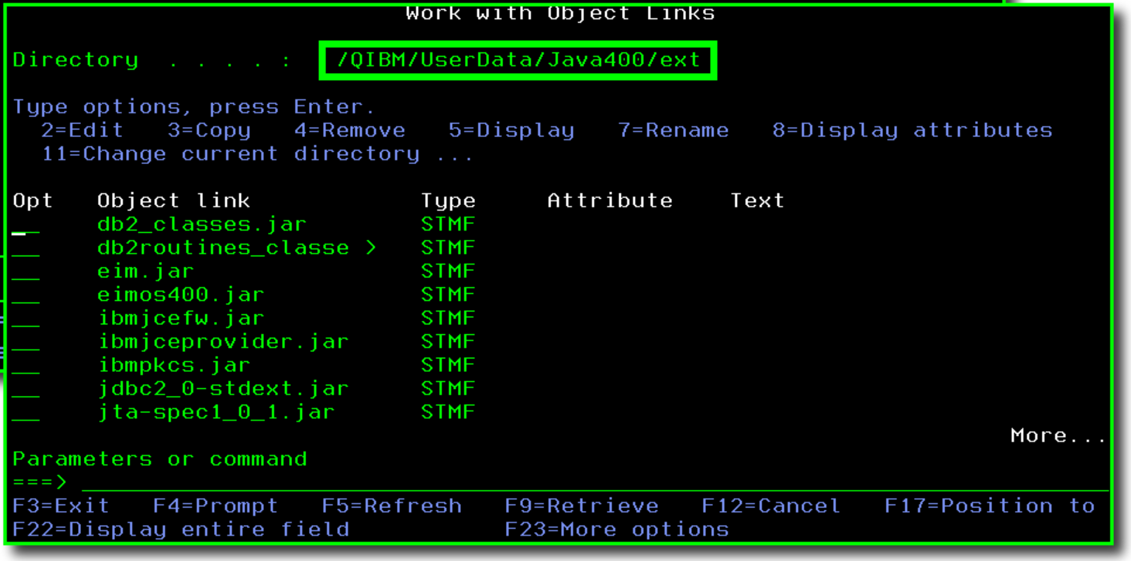
```
   // Set the publicly accessible field values
   // using JNI, then retrieve them.
   setIntField(JNIEnv_P:dim:widthId:1);
   setIntField(JNIEnv_P:dim:heightId:2);
   widthAfter = getIntField(JNIEnv_P:dim:widthId);
   heightAfter = getIntField(JNIEnv_P:dim:heightId);
   // Display the results
   displayString = 'Before: '
          + 'Width = '
          + %trim(%editc(widthBefore:'3'))
          + ' Height = '
          + %trim(%editc(heightBefore:'3'));
   dsply displayString;
   displayString = 'After: '
          + 'Width = '
          + %trim(%editc(widthAfter:'3'))
          + ' Height = '
          + %trim(%editc(heightAfter:'3'));
   dsply displayString;
   // Clean Up
   Air_closeConverter(cd);
   freeLocalRef(dim);
   freeLocalRef(dimClass);
   *inlr = *ON;
/end-free
```

External Jars

Locations on the IFS

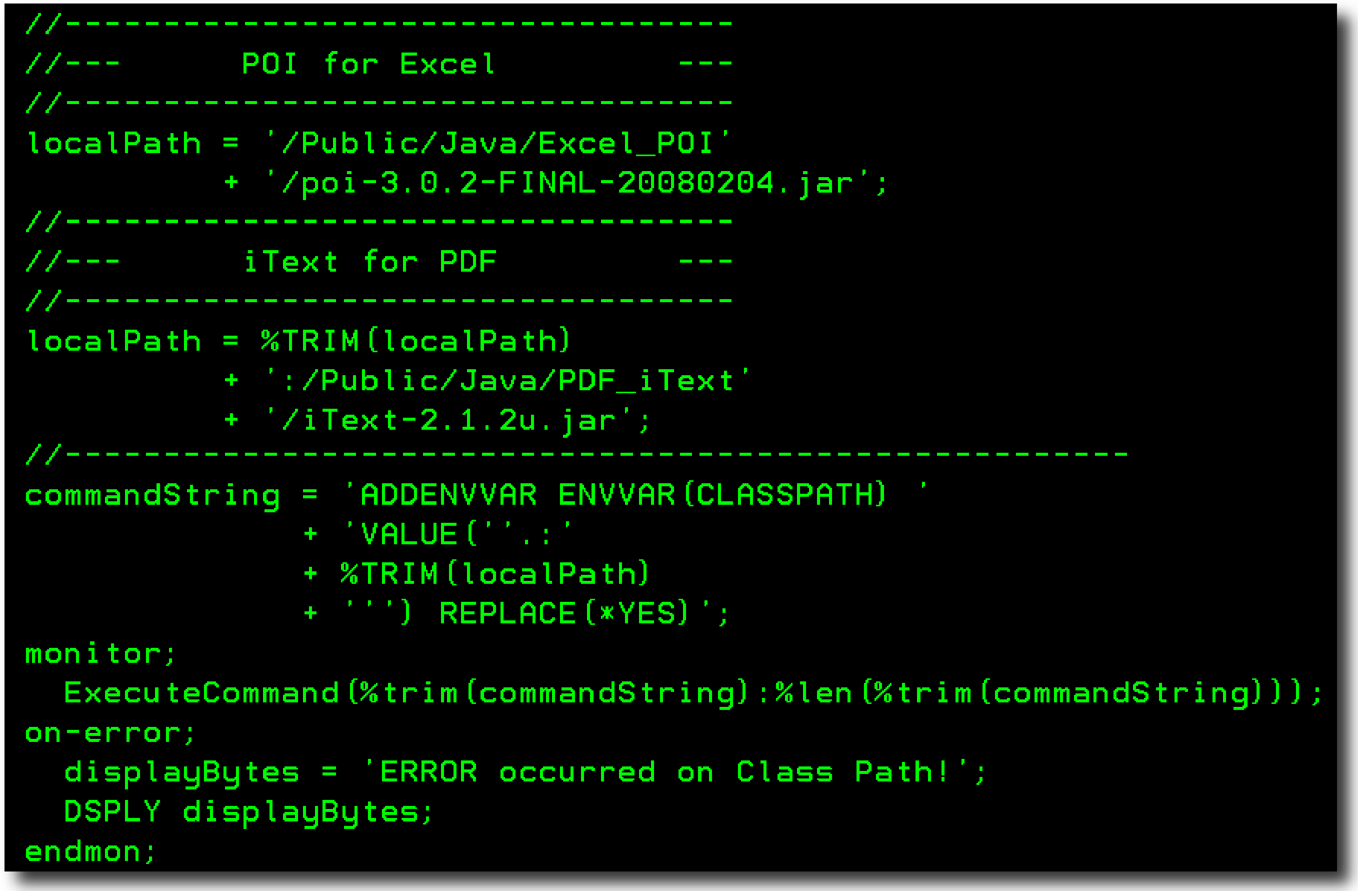


```
                         Work with Object Links

Directory  . . . . :   /Public/Java/PDF_iText

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link             Type      Attribute     Text
__    iText-2.1.2u.jar        STMF

Parameters or
===> ______
F3=Exit    F4=
F22=Display e
```



```
                         Work with Object Links

Directory  . . . . :   /QIBM/UserData/Java400/ext

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link             Type      Attribute     Text
__    db2_classes.jar         STMF
__    db2routines_classe >    STMF
__    eim.jar                 STMF
__    eimos400.jar            STMF
__    ibmjcefw.jar            STMF
__    ibmjceprovider.jar      STMF
__    ibmpkcs.jar             STMF
__    jdbc2_0-stdext.jar      STMF
__    jta-spec1_0_1.jar       STMF
                                                                    More...
Parameters or command
===> ________________________________________________________________
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field           F23=More options
```

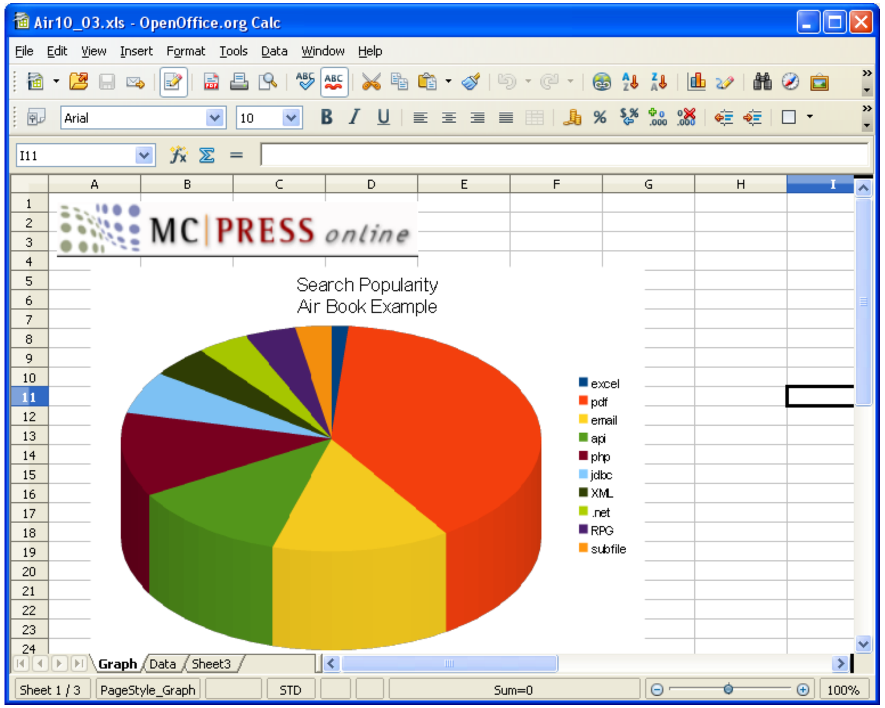
Setting the Class Path



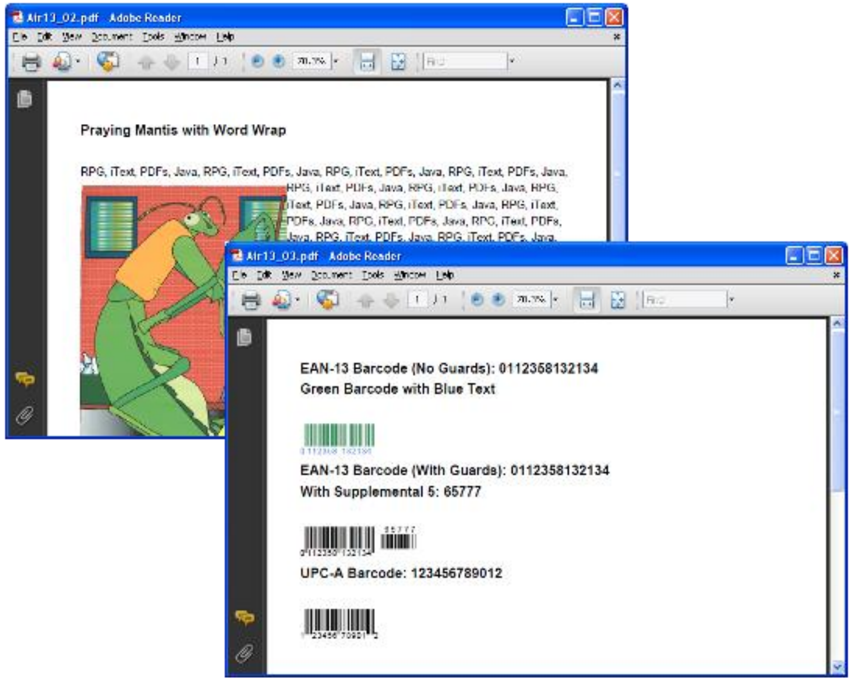
```
//----------------------------------
//---      POI for Excel         ---
//----------------------------------
localPath = '/Public/Java/Excel_POI'
          + '/poi-3.0.2-FINAL-20080204.jar';
//----------------------------------
//---      iText for PDF         ---
//----------------------------------
localPath = %TRIM(localPath)
          + ':/Public/Java/PDF_iText'
          + '/iText-2.1.2u.jar';
//------------------------------------------------------
commandString = 'ADDENVVAR ENVVAR(CLASSPATH) '
              + 'VALUE(''.:'
              + %TRIM(localPath)
              + ''') REPLACE(*YES)';
monitor;
  ExecuteCommand(%trim(commandString):%len(%trim(commandString)));
on-error;
  displayBytes = 'ERROR occurred on Class Path!';
  DSPLY displayBytes;
endmon;
```

Excel Spreadsheets with POI

- Create Service Program
- Create and Modify Excel Spreadsheets using RPG
- Formatting and Formulas
- Create Graphs and Charts



PDFs using iText


Air13_02.pdf Adobe Reader
File Edit View Document Tools Window Help
Air13_03.pdf Adobe Reader
File Edit View Document Tools Window Help

Praying Mantis with Word Wrap

RPG, iText, PDFs, Java, RPG, iText, PDFs, Java, RPG, iText, PDFs, Java, RPG, iText, PDFs, Java,
RPG, iText, PDFs, Java, RPG, iText, PDFs, Java, RPG,
iText, PDFs, Java, RPG, iText, PDFs, Java, RPG, iText,
PDFs, Java, RPG, iText, PDFs, Java, RPG, iText, PDFs,
Java, RPG, iText, PDFs, Java, RPG, iText, PDFs, Java,

EAN-13 Barcode (No Guards): 0112358132134
Green Barcode with Blue Text

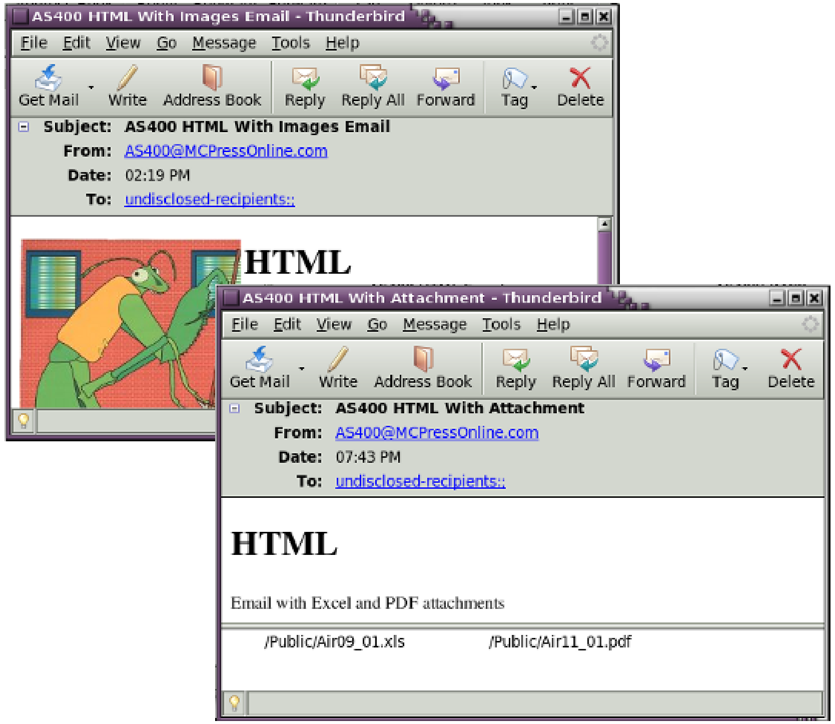
EAN-13 Barcode (With Guards): 0112358132134
With Supplemental 5: 65777

UPC-A Barcode: 123456789012

- Create Service Program
- Create and Modify PDFs using RPG
- Formatting, Tables and Links
- Barcodes

Send Email using JavaMail

- Send Email Directly from RPG
- Create HTML Formatted Email
- Attach Electronic Documents
- Embed Images into your Email



AS400 HTML With Images Email - Thunderbird
File Edit View Go Message Tools Help
Get Mail Write Address Book Reply Reply All Forward Tag Delete
AS400 HTML With Attachment - Thunderbird
File Edit View Go Message Tools Help
Get Mail Write Address Book Reply Reply All Forward Tag Delete

Subject: AS400 HTML With Images Email
From: AS400@MCPressOnline.com
Date: 02:19 PM
To: undisclosed-recipients:;

HTML

Subject: AS400 HTML With Attachment
From: AS400@MCPressOnline.com
Date: 07:43 PM
To: undisclosed-recipients:;

HTML

Email with Excel and PDF attachments

/Public/Air09_01.xls /Public/Air11_01.pdf

Advanced Integrated RPG

10% Discount Code: **OMNI2010**

Valid Through November 9th

http://www.mc-store.com/