

SDD ++: Extending SDD Capabilities

Sandeep Juneja
SAS Institute Inc., Cary, NC ,USA

Daniel Christen
SAS Institute AG., Switzerland

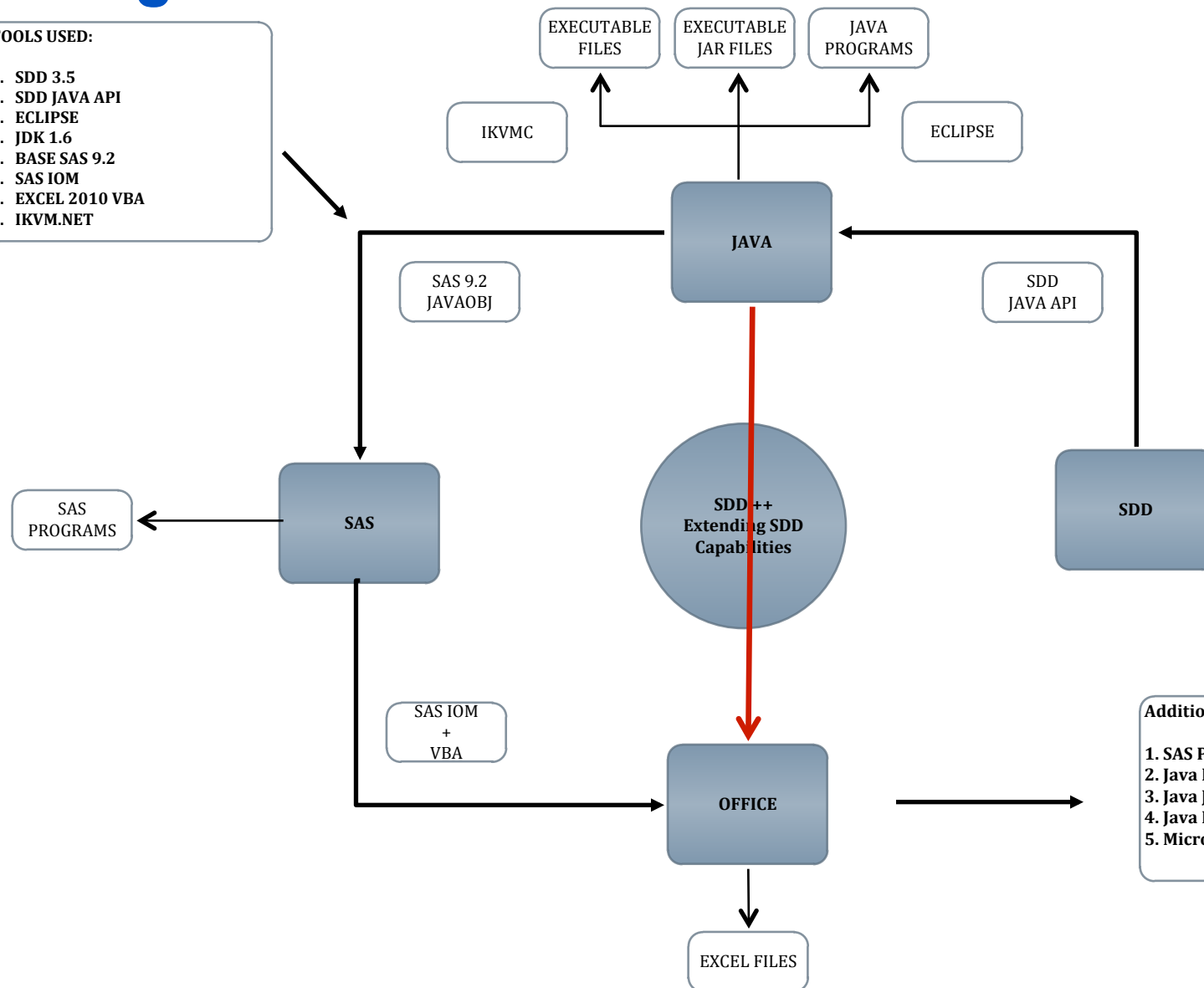


THE
POWER
TO KNOW®

Agenda

TOOLS USED:

1. SDD 3.5
2. SDD JAVA API
3. ECLIPSE
4. JDK 1.6
5. BASE SAS 9.2
6. SAS IOM
7. EXCEL 2010 VBA
8. IKVM.NET



Additional Capabilities:

1. SAS Programs / Macros
2. Java Programs
3. Java JARs
4. Java Executable exe files
5. Microsoft Excel Graphical capabilities

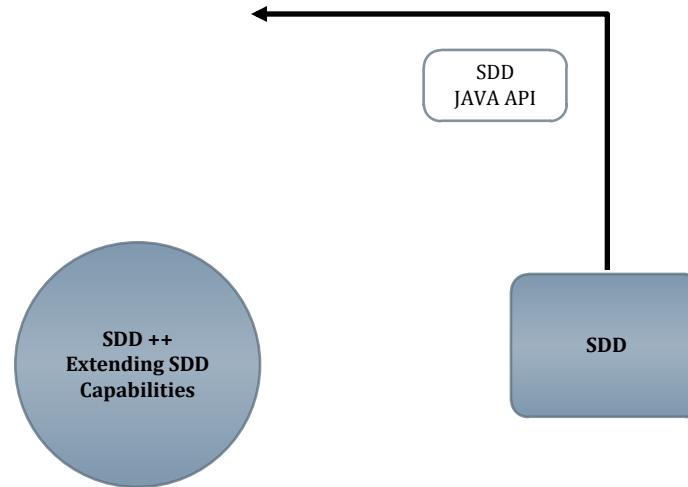
Getting Started – Software

- SAS Drug Development 3.x
- Java Development Kit:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Eclipse IDE for Java Developers:
http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/R/eclipse-java-indigo-win32-x86_64.zip
- SDD API:
<ftp://ftp.sas.com/techsup/download/hotfix/35drugdev.html>
- IKVM.net: <http://sourceforge.net/projects/ikvm/files/>

Getting Started – User guides

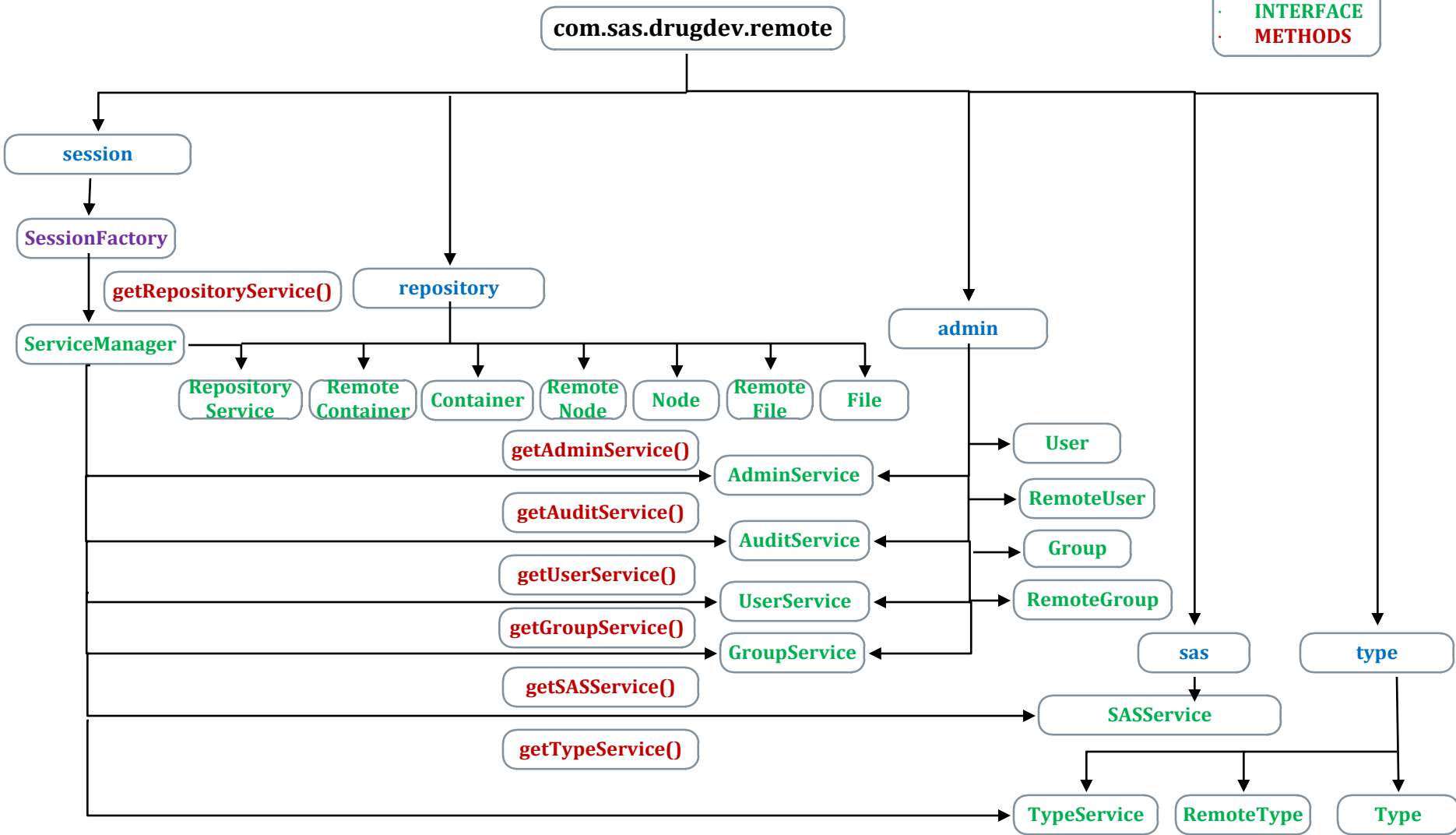
- SDD Remote API User guide:
ftp://ftp.sas.com/techsup/download/hotfix/drugdev/35drgmacro02/SDD_Macros.pdf
- Eclipse Video Tutorial:
<http://sourceforge.net/projects/eclipsetutorial/files/1.%20Total%20Beginners/Version%201.0/>
- SAS JavaObj:
<http://support.sas.com/rnd/base/datastep/dot/javaobj.html>
- SAS IOM:
<http://support.sas.com/rnd/itech/doc/dist-obj/index.html>
- VBA IOM code:
<http://support.sas.com/rnd/itech/doc/dist-obj/winclnt/wkspmgr/samples.html>

SDD Java API



SDD Java API Overview

- PACKAGES
- CLASSES
- INTERFACE
- METHODS



Establishing SDD Connection via API

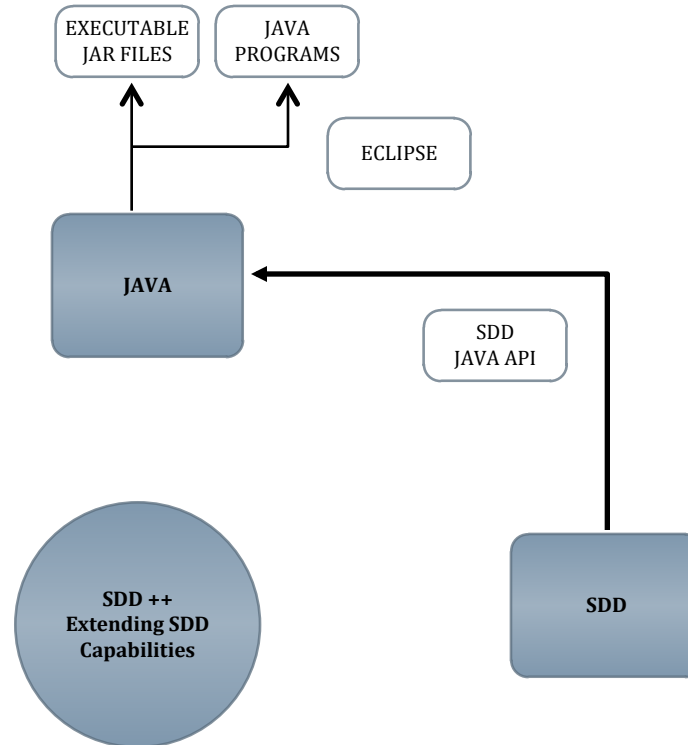
```
session=SessionFactory.newSession(  
new URL ( "https://sdd<instance>.sas.com/sddremote"),  
new UsernamePasswordCredentials("<userid>", "<password>".toCharArray()));
```

```
ServiceManager serviceManager = session.getServiceManager();
```

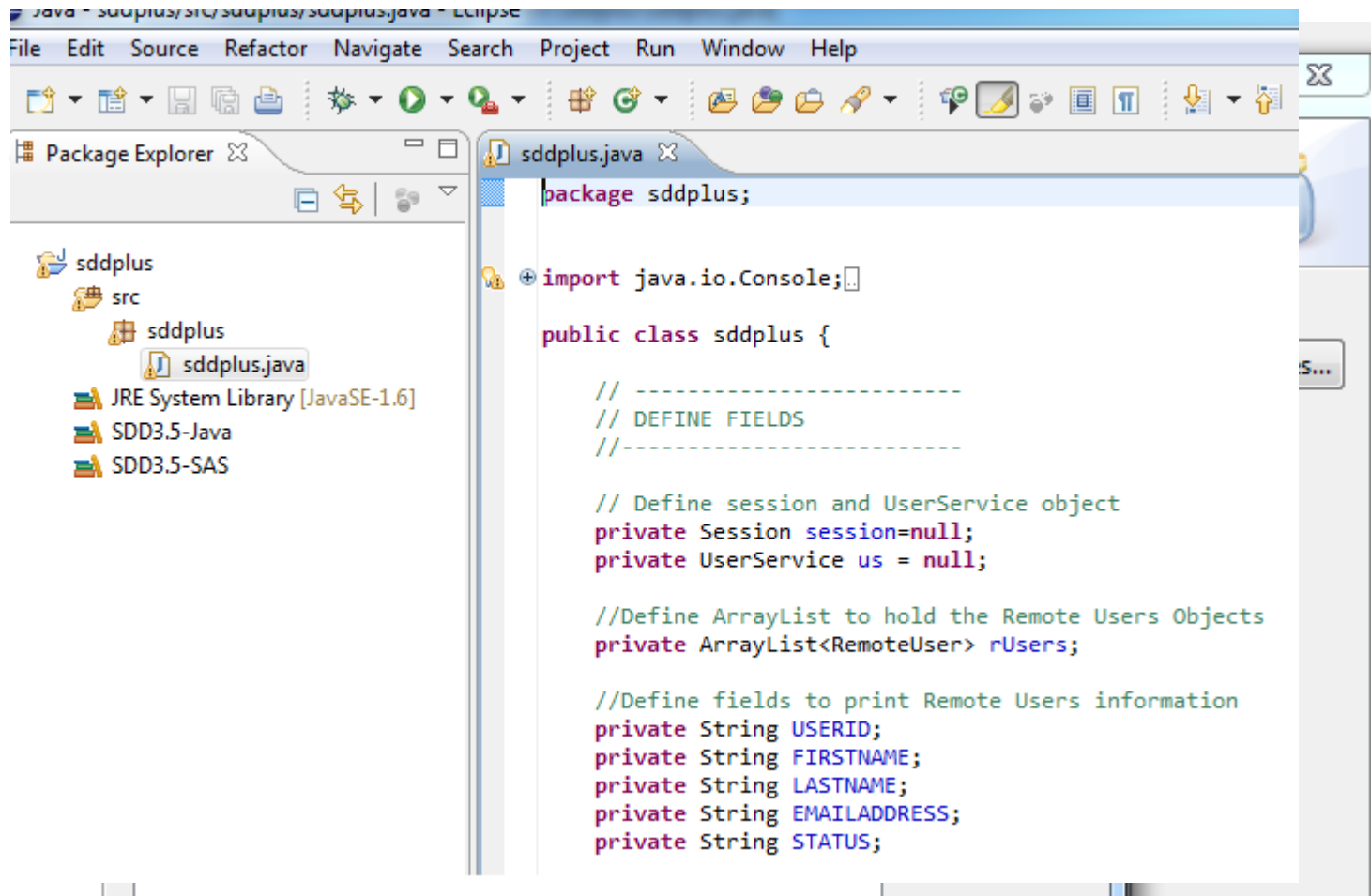
```
UserService us = serviceManager.getUserService();  
RepositoryService rs = serviceManager.getRepositoryService();  
GroupService gs = serviceManager.getGroupService();  
SASService ss = serviceManager.getSASService();  
TypeService ts = serviceManager.getTypeService();
```

```
AdminService as = serviceManager.getAdminService();  
AuditService aus = serviceManager.getAuditService();
```

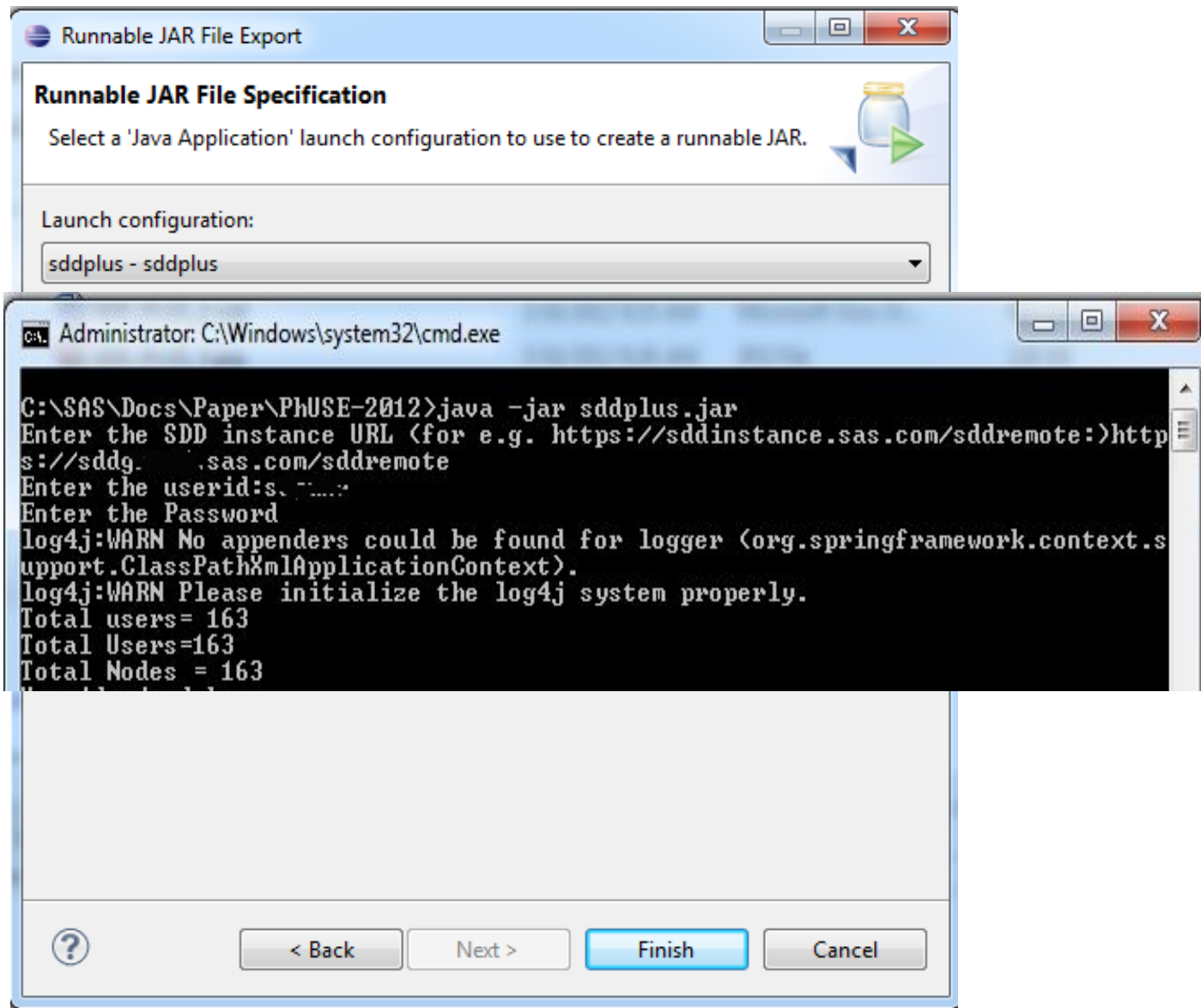
SDD - Java



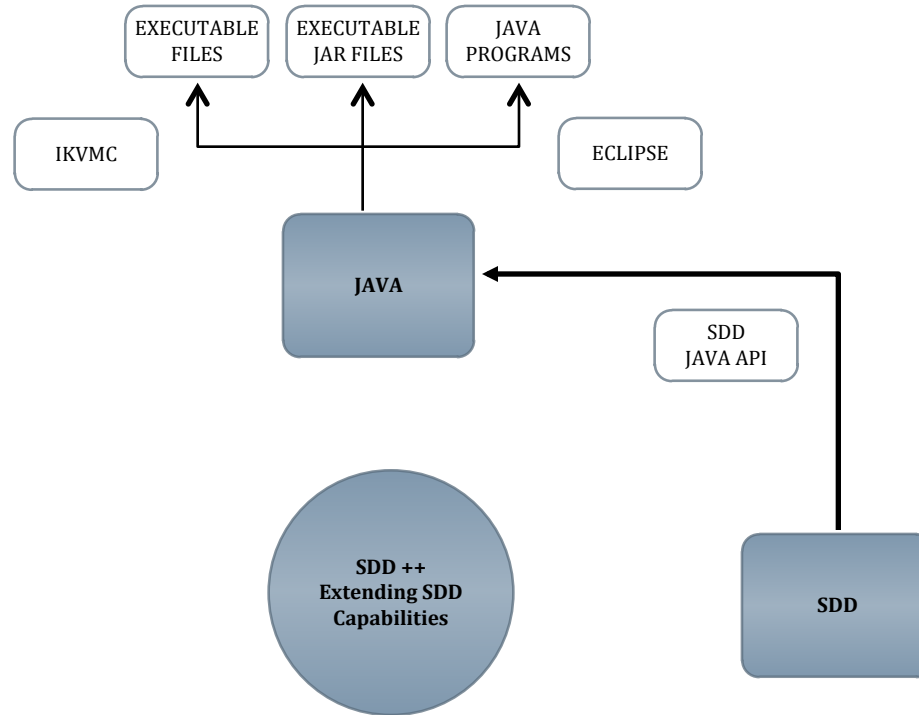
Setting Eclipse Project



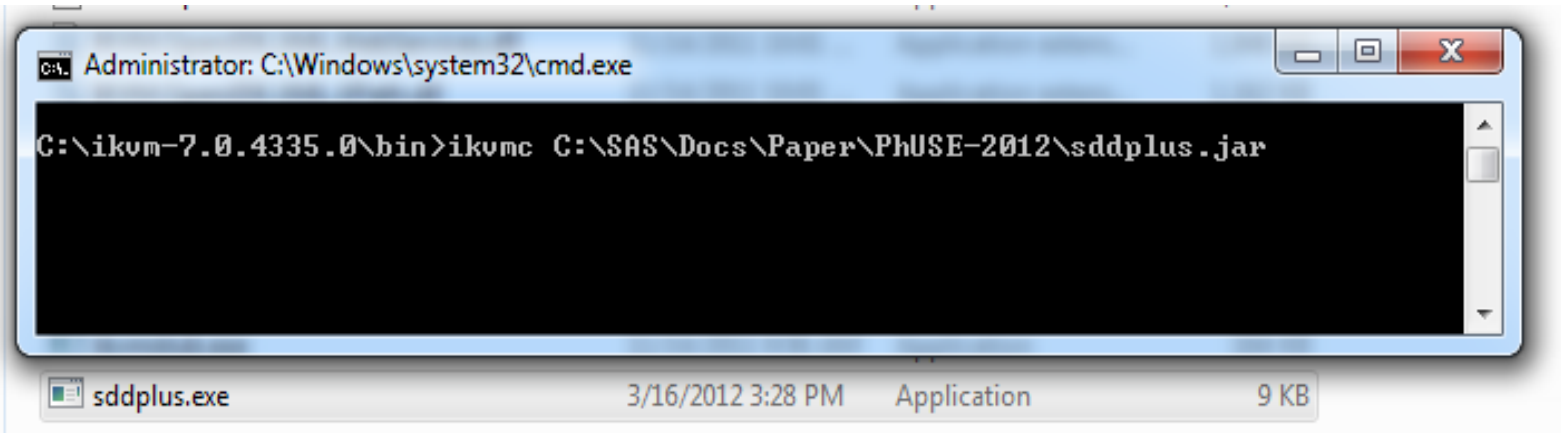
Generating and Executing Jar file



Convert jars to exe



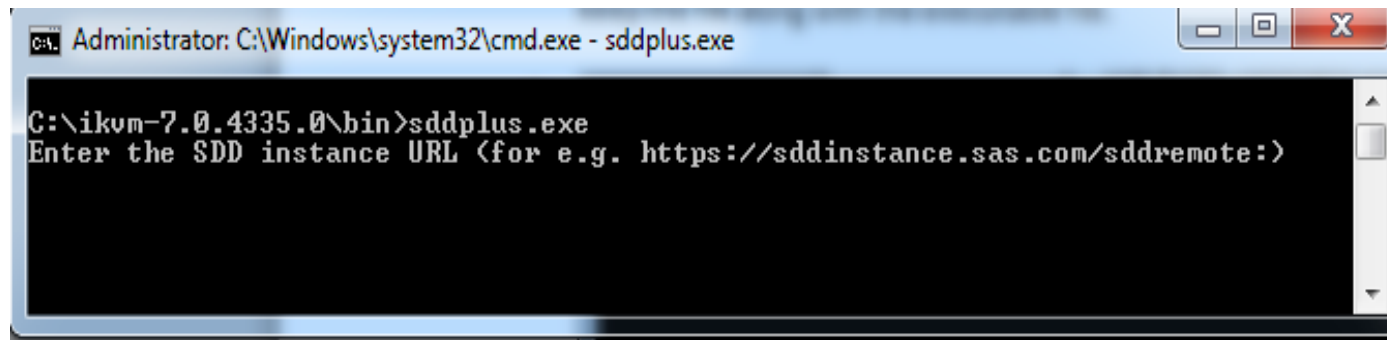
Convert jar to Executable (exe) file



A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The command prompt shows the command `ikvmc C:\SAS\Docs\Paper\PhUSE-2012\sddplus.jar` being executed. Below the command prompt, a taskbar entry for `sddplus.exe` is visible, showing the file icon, name, date and time (3/16/2012 3:28 PM), type (Application), and size (9 KB).

```
C:\ikvm-7.0.4335.0\bin>ikvmc C:\SAS\Docs\Paper\PhUSE-2012\sddplus.jar
```

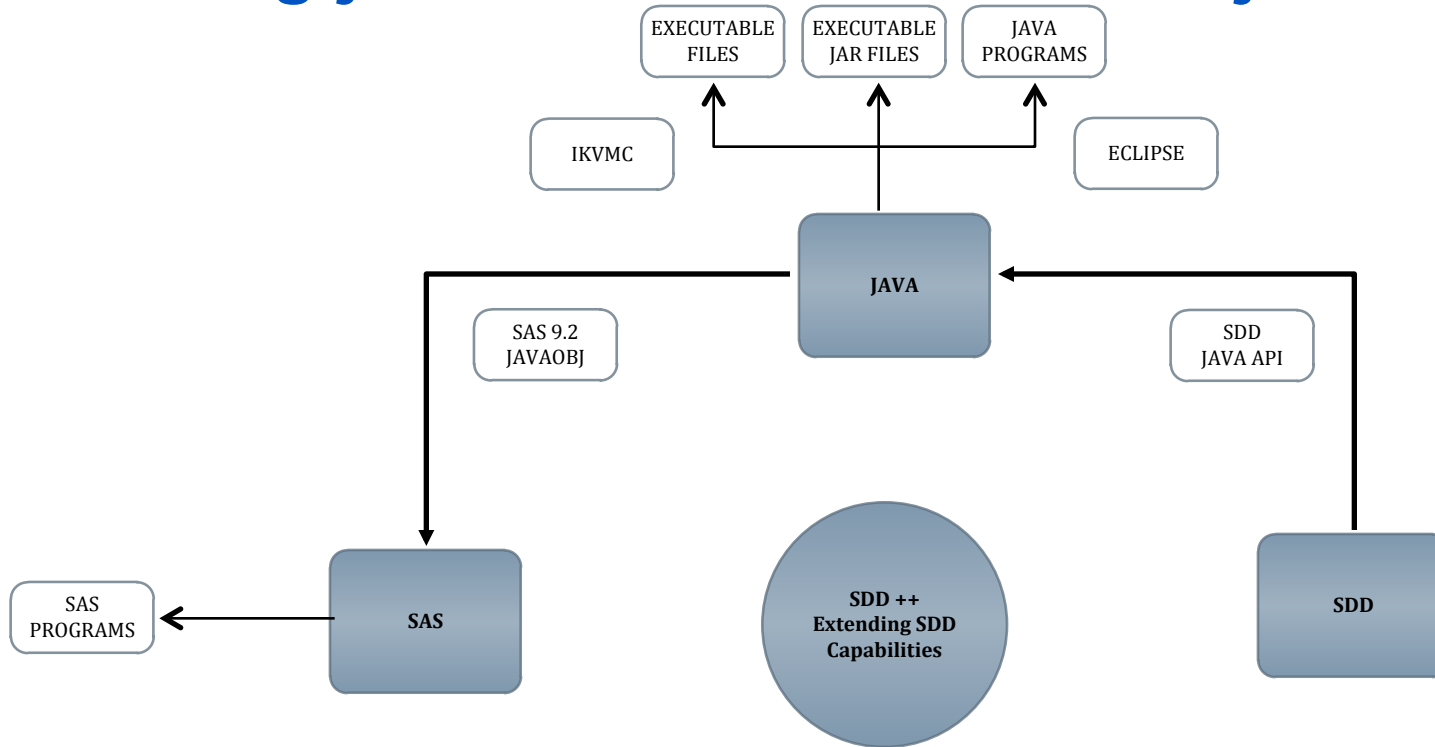
sddplus.exe 3/16/2012 3:28 PM Application 9 KB



A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe - sddplus.exe". The command prompt shows the command `sddplus.exe` being executed. The output of the command is "Enter the SDD instance URL <for e.g. https://sddinstance.sas.com/sddremote:>".

```
C:\ikvm-7.0.4335.0\bin>sddplus.exe
Enter the SDD instance URL <for e.g. https://sddinstance.sas.com/sddremote:>
```

Using jar files via SAS JavaObj



SAS JavaObj

- Javaobj provides a mechanism for instantiating Java classes, and accessing fields and methods on the resultant objects.

```
declare javaobj sdd("sddplus/sddplus");
```

```
sdd.callVoidMethod  
("connectSDD",&sddurl,&sdduser,&sddusrpwd");
```

```
sdd.callIntMethod("getCnt",recCnt);
```

```
sdd.delete();
```

100%



```
\RAPI\API\3.5\SAS\lib:C:\SDD\jars;
```

macro"

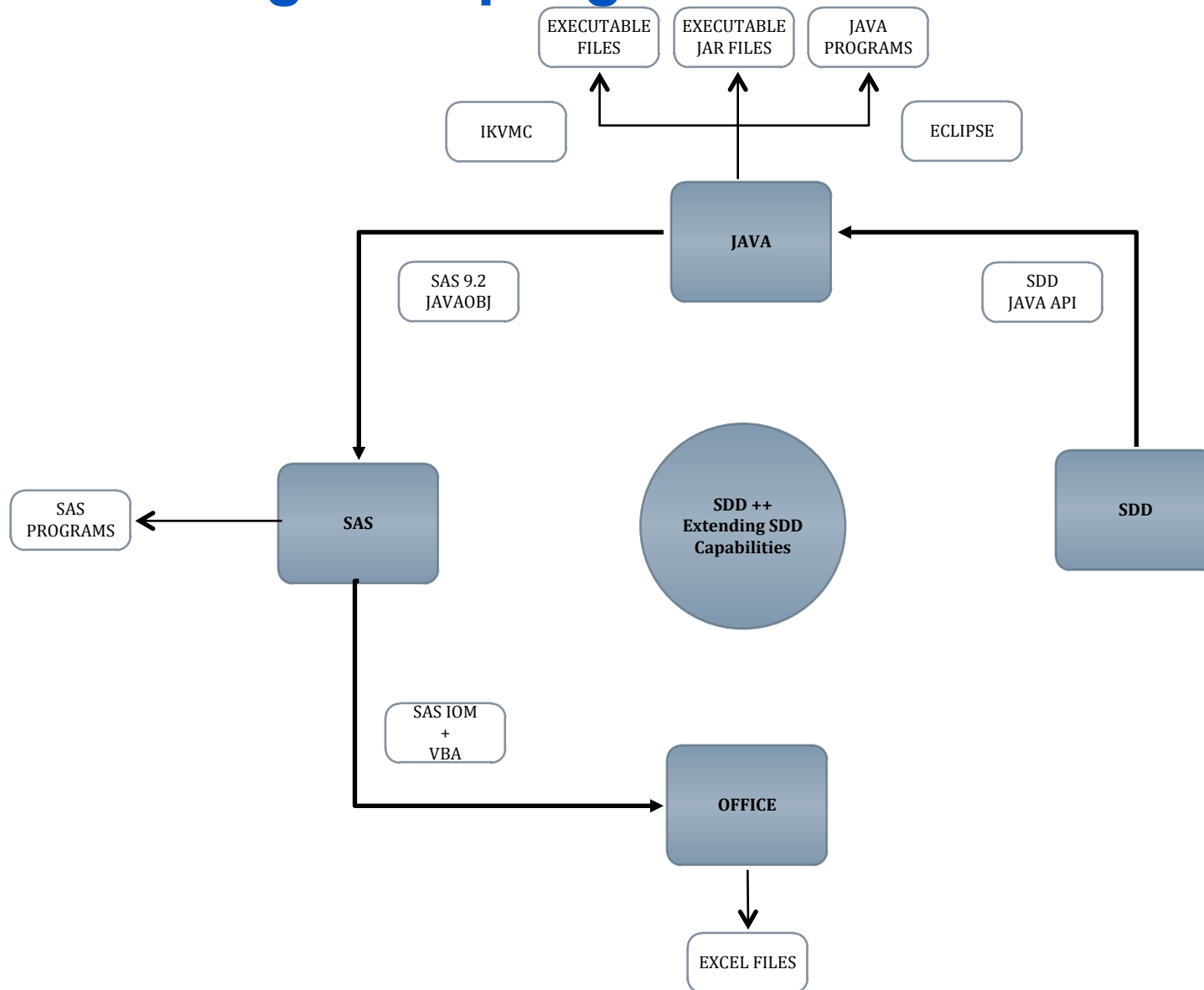
Call Jar file via SAS JavaObj using PC SAS

The screenshot displays the SAS software interface. The main window shows a data table with the following columns: **userid**, **firstname**, **lastname**, **emailaddress**, and **status**. The data is organized into rows, with the first row (row 8) showing a user with **userid** 'regUs-1331750803751', **firstname** 'First Name', **lastname** 'Last Name', **emailaddress** 'test@test.com', and **status** 'active'. Subsequent rows (9-24) show users with **userid** values starting with 'new' and similar **firstname**, **lastname**, **emailaddress**, and **status** information.

The interface includes a menu bar (File, Edit, View, Tools, Data, Solutions, Window, Help), a toolbar, and a sidebar with an Explorer pane showing the contents of the 'Work' directory, including folders like 'Sasmacr' and 'Sddusers'. The bottom status bar shows the current file is 'sddplus.sas'.

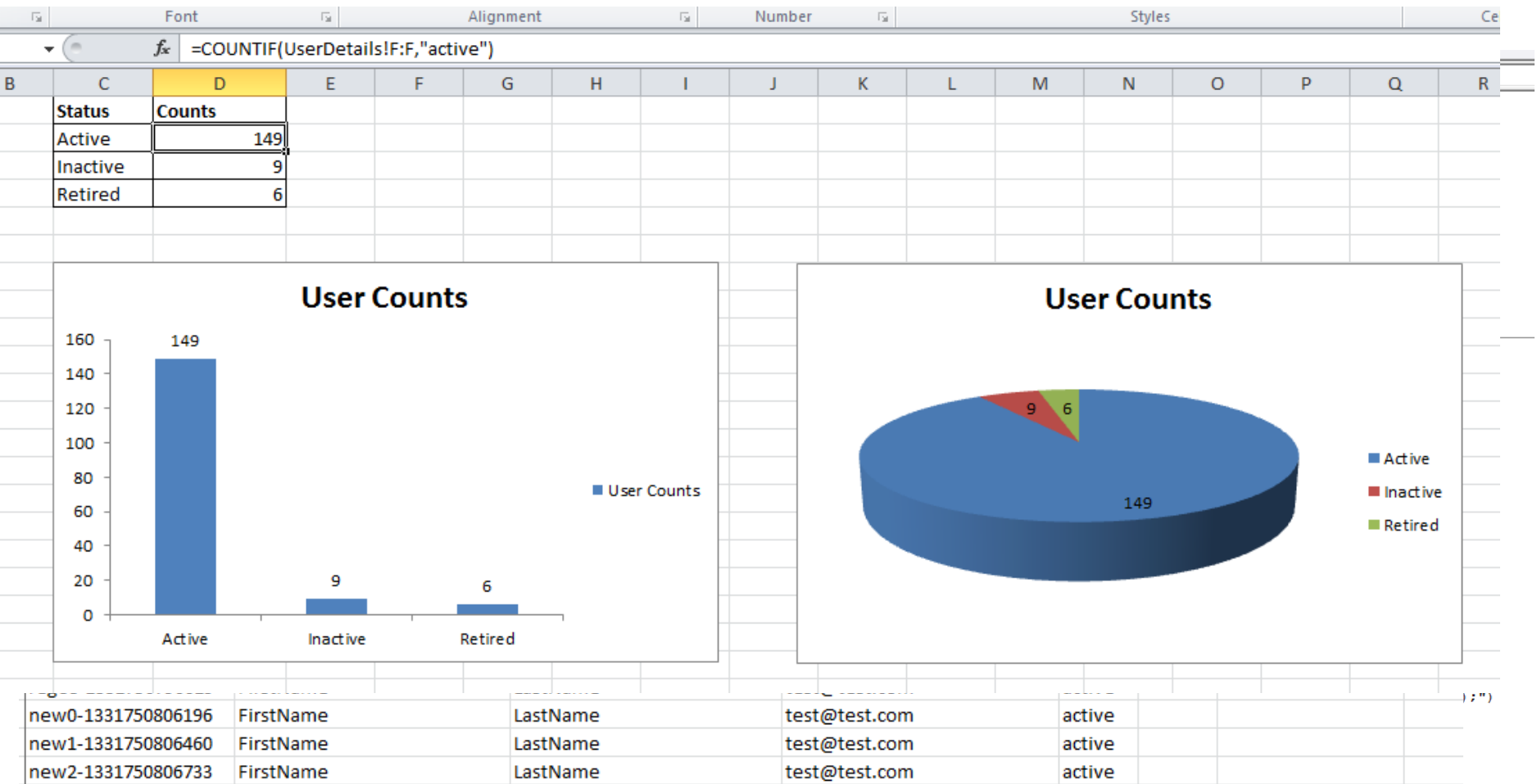
	userid	firstname	lastname	emailaddress	status
8	regUs-1331750803751	First Name	Last Name	test@test.com	active
9	new21-1331750813053	First Name	Last Name	test@test.com	active
10	new22-1331750813226	First Name	Last Name	test@test.com	active
11	new23-1331750813439	First Name	Last Name	test@test.com	active
12	new24-1331750813611	First Name	Last Name	test@test.com	active
13	new25-1331750813785	First Name	Last Name	test@test.com	active
14	new26-1331750814198	First Name	Last Name	test@test.com	active
15	new27-1331750814377	First Name	Last Name	test@test.com	active
16	new28-1331750814544	First Name	Last Name	test@test.com	active
17	new29-1331750814701	First Name	Last Name	test@test.com	active
18	new30-1331750814865	First Name	Last Name	test@test.com	active
19	new31-1331750815036	First Name	Last Name	test@test.com	active
20	new32-1331750815205	First Name	Last Name	test@test.com	active
21	new33-1331750815368	First Name	Last Name	test@test.com	active
22	new34-1331750815523	First Name	Last Name	test@test.com	active
23	new35-1331750815680	First Name	Last Name	test@test.com	active
24	new36-1331750815852	First Name	Last Name	test@test.com	active

Calling SAS programs via VBA



SAS IOM

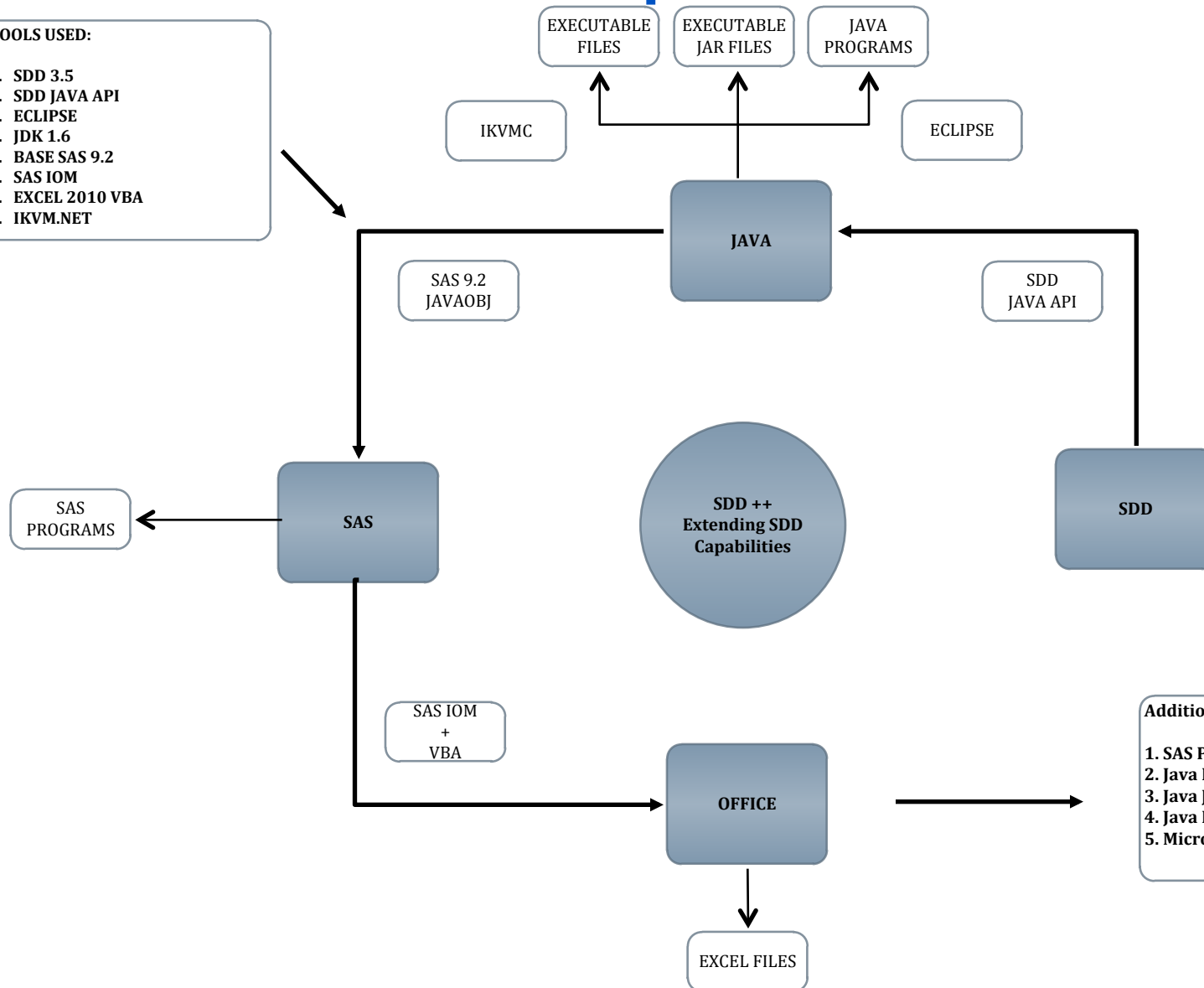
- The [Integrated Object Model](#) provides distributed object interfaces for conventional SAS features.
- It enables you to develop component-based applications that integrate SAS features into your enterprise application.
- The principal interfaces of IOM include:
 - **Workspace** (a SAS session)
 - **LanguageService** (submit, list/log, stored processes)
 - **DataService** (librefs, plus access through ADO, OLE-DB or JDBC)
 - **FileService** (filerefs and directories)
 - **Utilities** (result packages, formats, host info)



Extended SDD Capabilities Benefits

TOOLS USED:

1. SDD 3.5
2. SDD JAVA API
3. ECLIPSE
4. JDK 1.6
5. BASE SAS 9.2
6. SAS IOM
7. EXCEL 2010 VBA
8. IKVM.NET



Additional Capabilities:

1. SAS Programs / Macros
2. Java Programs
3. Java JARs
4. Java Executable exe files
5. Microsoft Excel Graphical capabilities

Questions





Email: Sandeep.Juneja@sas.com
Daniel.Christen@sas.com

SDD Forum: <https://communities.sas.com/community/support-communities/sas-drug-development>

Thank you

www.sas.com