The Use of Java with The SAS® System

Clare Nicklin
Amadeus Software Limited
20th May 2003
Introduction

• AppDev Studio™ is SAS’ flagship Java development tool

• But it is not the only route to SAS from Java

• Four potential routes in to SAS are discussed

• Including their software requirements, methods, advantages, and disadvantages
Client- or Server-Side Processing

• **Client**
  – Java applications, applets, and additional classes
  – Run from the client machine (or browser) in a distributed network
  – Advantages: usable, interactive, rich-client interfaces
  – Restrictions: downloading Java classes to client machine

• **Server**
  – Servlets, JavaServer Pages (JSPs), and additional classes
  – Run from the server machine (or web server) in a distributed network
  – Advantages: quick and functional, downloads small files to client machine
  – Restrictions: less interactive, thin-client interfaces restricted to the look and feel of HTML
Java Access To SAS

- AppDev Studio - webAF™
- Directly Using the IOM Server - SAS® Integration Technologies
- Java Database Connectivity (JDBC)
- SAS Batch Processing with Remote Method Invocation (RMI)
• **Probable Software Requirements**
  – AppDev Studio
  – SAS/Connect® or Integration Technologies
  – Additional software:
    • Java Development Kit (JDK)
    • Web server, such as Apache
    • Servlet engine, such as Tomcat

• **Method**
  – Supports client- and server- side development
  – Integrated visual programming environment
    • Wizards
    • Drag and drop
  – Allows simple communication with the SAS System
    • Submission of Base SAS code and utilisation of remote SCL classes
  – webAF provides a set of JavaBeans called ‘TransformationBeans’
Advantages
- Simple and effective for quick development
- Can decrease the time and resources needed to develop web-based applications to interface with SAS
- Many of the complicated tasks involved in connecting to SAS or manipulating data are hidden from the developer

Disadvantages
- All the software involved can be expensive
- Complicated deployment process
- Benefits have to be compared against the costs in deciding on AppDev Studio’s utility
• Probable Software Requirements
  – Integration Technologies and the classes it provides
    • IOM Spawner listens constantly for calls to SAS
  – Additional software:
    • JDK
    • Web server
    • Servlet engine (for server-side development)

• Method
  – Java programs make use of the SAS IOM server almost as if it were a set of Java objects
  – JAR files on the Client Side Components Disk
Integration Technologies(2)

- **Advantages**
  - JAR files on the Client Side Components Disk, without having to license software such as AppDev Studio

- **Disadvantages**
  - Development is quicker in webAF using pre-existing classes
  - Several more JAR files accompany AppDev Studio
• Probable Software Requirements
  – SAS Integration Technologies or SAS/SHARE®
  – Additional software:
    • JDK
    • Web server
    • Servlet engine (for server-side development)

• Method
  – Connect to, update, and query databases using the Structured Query Language (SQL)
  – Standard SQL statements such as SELECT, CREATE, UPDATE, INSERT, and DELETE are supported
  – Drivers for SAS can be found attached to other software such as SAS/SHARE
  – The SAS/SHARE driver for JDBC enables applications to access and update any SAS data that is available through SAS/SHARE, from Java programs
Advantages
- JDBC usage can be relatively straightforward
- SAS users often already have access to the necessary software
- They also often already have knowledge of SQL
- Industry standard, examples of its use can be relatively easy to find.

Disadvantages
- Drivers for SAS are not freely accessible
- Wrapped up with other potentially expensive software
SAS Batch Processing with RMI(1)

• Probable Software Requirements
  – The SAS System on a server machine alone would be sufficient to use RMI technology for batch processing SAS programs
  – A JDK needs to be installed on both the machines aiming to communicate
  – Web server

• Method
  – RMI is a standard used to communicate between two Java Virtual Machines via network protocols such as TCP/IP
  – Java application running on a local machine collects information from a user using a graphical user interface, sends that information to a server for processing, which then returns a response
SAS Batch Processing with RMI(2)

• Advantages
  – Standard Java technology
  – Supplied with any recent JDK
  – SAS programs can be run in batch
  – Server based classes access results and can send them back to the client as sets of objects

• Disadvantages
  – Can take time to research and set up correctly
  – Several classes involved to establish the necessary connections between the client and server machines
  – Process can be a little time-consuming to begin with
  – Example in paper would not be recommended for large amounts of data, passing large sets of results back to the client could be resource intensive
Other Routes

- **SAS/IntrNet® or other CGI technologies**

- **Web services**
  - Platform independent
  - Based on a set of industry standards which provide basic functions via Internet protocols
  - Relatively inexpensive

- **Methods can be used collectively**
  - RMI technology could be used to run SAS code to manipulate data and create results data sets while JDBC could then be used to access and retrieve those results
  - AppDev Studio in fact utilises technologies such as JDBC and RMI to build much of its operability
Conclusion

• Specialist tools such as AppDev Studio are not the only way to use SAS from Java based applications
• Other routes provided by SAS
• Java standards that can be adapted which can utilise SAS
• It is important to note that there are alternatives, even if some of these may use SAS in more simplistic or roundabout ways
Questions?

Clare Nicklin
Clare.Nicklin@amadeus.co.uk

Amadeus Software Ltd
www.amadeus.co.uk