

Porting Smart Cards to AIX



By George Kraft IV

Smart card standards have evolved as plastic card technology has begun using microprocessor silicon chips and programmable memory. This article presents an overview of smart card technology and applications.

Today, plastic cards used as “credit cards” have embossed customer name and account number, signature strip, magnetic strip, bar code strip, hologram, and in some cases, the customer’s photograph. Smart card technology introduces microprocessor silicon chips and programmable memory to the already existing plastic card standards. Figure 1 shows a sample smart card.

Smart cards can electronically hold credit card account information, health insurance policies, driver’s license, identification badge, or any other form of card information normally carried in a wallet. In addition, smart cards can function as an electronic purse to hold electronic cash. Electronic cash, or *e-cash*, can store an electronic representation of money that can be used for small miscellaneous expenditures such as vending machine purchases, bus rides, and metered parking.

Smart Card Chip

The smart card chip, about five millimeters in size, is embedded in the upper left corner

of the plastic card. Outside the card, on the surface, is the contact plate for serial input and output (I/O) to the card reader. Cards without this contact are smart cards with an embedded antenna that usually operates within the one to fifteen centimeter range.

The chip generally has a central processing unit; EPROM, to hold its operating system; RAM; and EEPROM for resident smart card applications and data, shown in Figure 2. The job of the operating system is to receive commands from the 9600 baud serial connection, access the applications and data on the card, and send a response back to the serial connection. The smart card is a plastic “credit card” with an embedded miniature computer.

The ISO 7816 standard defines the physical characteristics of the smart card, including the I/O. This standard defines a consistent means for communicating with smart cards manufactured by different vendors. However, different cards and card manufacturers have distinguishing features.

The embedded operating systems represent the primary differences between smart cards. For this reason, smart-card developer toolkits are provided to create card-resident applications and data. This information usually “initializes” the cards for a particular application during manufacturing.¹



George Kraft IV

¹ Visit <http://scs.bs.boeblingen.ibm.com/> to see IBM’s smart card toolkit designed to support the IBM MultiFunction Card (MFC) and other card operating systems.



Figure 1. Physical characteristics of the smart card

OpenCard Framework

The OpenCard Framework (OCF) consortium held its inaugural meeting at the CardTech/SecureTech '98 trade show in Washington, D.C. during April 1998. At this meeting the consortium announced the release of the OCF Version 1.0 application programming interface (API) written in Java™. OCF was developed for host-based Java applications to be smart card aware and to be able to drive smart-card base applications. OCF provides smart-card reader device and smart-card operating system independence through Card Terminal and Card Services respectively.²

Host-resident, smart card-aware Java applications using OCF can be transported between host operating systems, smart-card terminal devices, and vendor-specific smart cards. To port OCF to particular platforms such as AIX®, a smart-card terminal device must be selected. Then a Java Native Interface (JNI) driver must be written to span between the host operating system and OCF running in the Java Virtual Machine (JVM). An example of this for AIX can be found in the *com/ibm/opencard/terminal/ibm5948/natives/aix* directory from the sample source downloaded from the OpenCard home page.

OCF provides a card extension service to discern smart card differences; for example, the Europay MasterCard Visa (EMV) e-cash services. OCF can call upon the registered

EMV extensions for the smart-card electronic purse. OCF's auxiliary extendibility makes it flexible enough to accommodate most vendor smart cards.

Major PC and smart card companies formed the Personal Computer/Smart Card (PC/SC) workgroup in May 1996. The PC/SC specification, based on the ISO 7816 standards, is designed to interface smart card readers and cards with PCs through device-independent APIs for smart card-aware applications on Windows™ and Windows NT™.³

OCF-compliant applications written in Java migrate easily to Windows, because OCF can use the PC/SC drivers for smart card readers; therefore, OCF and PC/SC smart card applications coexist on the WinTel platform. OCF smart card applications running on various platforms can easily migrate to Windows; however, PC/SC applications cannot migrate away from Windows and Windows NT. See URL <http://www.opencard.org/misc/OCF-FAQ.html#PCSC> for more information regarding OCF and PC/SC compatibility.

The Java Card

We have talked about ISO 7816 physical characteristics, the OCF and PC/SC smart card-aware host applications, and various smart-card operating systems with supporting developer's toolkits to create card-resident, smart card applications.

² Visit <http://www.opencard.org/> for demos and more information regarding OCF.

³ See URL <http://www.smartcardsys.com/> for more information regarding PC/SC.

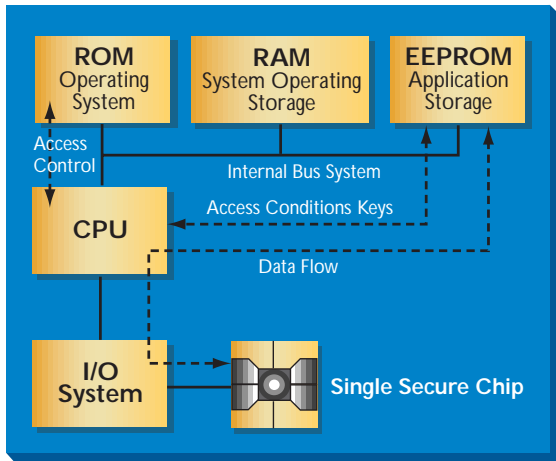


Figure 2. The smart card chip

The Java Card™, one such smart-card operating system, can run Java applications on the card.

The Java Card framework defines a reduced set of APIs for developing smart card-resident applications. Missing from the Java Card are dynamic class loading, security manager, threads and synchronization, object cloning, finalization, and large primitive datatypes (float, double, long, and char). However, the Java Card is not a stripped down Java Development Kit (JDK). Unlike the JVM on a PC, the Java Card Virtual Machine runs continuously. When the Java Card is removed from the reader and powered down, the information on the card is persistently stored in the

card's EEPROM.⁴ Developers must handle garbage collection manually, because the Java Card Virtual Machine does not.

Conclusion

This brief introduction to smart cards shows some advantages of programming smart card-aware applications and smart card-resident applications in Java using OCF and Java Card respectively. Although this article is not announcing any specific smart-card product releases for the AIX operating system on the RS/6000™, it suggests that a new "open" technology is ready to be utilized.

The OpenCard Framework has released the OCF Version 1.0 API for developers to create smart card-aware Java applications, and JavaSoft™ has released Java Card Version 2.0 API for developers to create smart card-resident Java applications.



George Kraft IV, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Kraft is an advisory software engineer for IBM's Network Computer Division. He recently moved from IBM's RS/6000 Division where he worked on the AIX integration of the IBM Network Station™. He has a BS in Computer Science and Mathematics from Purdue University.

⁴ See URL <http://www.javasoft.com/javacard/> for more information regarding the Java Card API for smart cards.