JCOP10 Technical Brief

Overview: *This document contains a simple overview about the technical capabilities of the first member of the JCOP card OS family, the DES-only version. Requests for further information may be directed at javacard@zurich.ibm.com.*

1. Basic specifications

JCOP is an IBM BlueZ implementation of the basic specifications [1] and [2] including refinements from Visa International set in the Visa OpenPlatform Card Implementation Guides (http://www.visa.com/nt/suppliers/vendor). All necessary clarifications from ISO7816 and EMV 2000 are also incorporated into the implementation where so required by [1] and [2].

JCOP10v1 is the first member of this family. It conforms to the VOP Card Implementation Guide 2.1.1 Compact from August 2000.

Its successor, JCOP10v2.0 is compliant with the Visa OpenPlatform Card Implementation Requirements Configuration 1, Version 2.0 from February 2002.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default][1]

ISO7816 T=0 direct convention[1]

ISO7816 T=1 inverse convention[1]

ISO7816 T=0 inverse convention[1]

2.2 Supported speeds

At the default clock rate of 3.57 MHz, the following communication speeds can be attained:

9600 bit/sec [default]

19200 bit/sec

38400 bit/sec

115200 bit/sec

[1] The contact protocols of JCOP can be configured to support the clock-stop feature of certain terminals to save power consumption (typically done in mobile phones). This feature is available since JCOP10v2.

3 Memory availability for applications

3.1 EEPROM

3.1.1 Persistent Java heap

Used for allocating persistent objects, applets, and storage of post-issuance loaded applet code (aka packages).

Size: 15kB/31kB* (no custom ROM applets).

3.1.2 Transaction buffer

Used to save data written transactional, e.g. all persistent byte and short stores, as well as persistent parameters to Util.arrayCopy; see [1].

Size: 512 bytes

3.2 RAM

3.2.1 Transient Java heap

Used for allocating transient objects and arrays of type CLEAR_ON_RESET and CLEAR_ON_DESELECT.

Size: 651 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communications data

Size: 261 bytes

3.2.3 Java stack

Used to hold call parameters, local variables, and stack frames of the VM.

Size: 200 bytes

3.3 ROM

16kB/64kB* free for applications

* Depending on hardware configurations (see section 5)

4 Supported optional features

Certain features listed in [1] and [2] are not defined to be mandatory. The ones implemented in JCOP are listed below.

4.1 JavaCard

4.1.1 Garbage Collection

Fully implemented: Deleted objects, applets, and packages are fully reclaimed and the space can be used for other purposes after deletion.

4.1.2 Cryptographic Algorithms

The following JavaCard API constants (see [1]) are implemented by this version of JCOP:

Ciphers:

ALG_DES_CBC_NOPAD

ALG_DES_CBC_ISO9797_M1

ALG_DES_CBC_ISO9797_M2

ALG_DES_ECB_NOPAD

ALG_DES_ECB_ISO9797_M1

ALG_DES_ECB_ISO9797_M2

Signatures:

ALG_DES_MAC8_NOPAD

ALG_DES_MAC8_ISO9797_M1

ALG_DES_MAC8_ISO9797_M2

RandomData:

ALG_SECURE_RANDOM

KeyTypes:

LENGTH_DES

LENGTH_DES3_2KEY

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

4.2 OpenPlatform

4.2.1 Global PIN

Fully implemented: All described APDU and API interfaces for this feature are present.

5 Supported Hardware

The identical JCOP mask can run on these platforms and automatically make use of their resources, i.e., create a bigger persistent Java heap, and make more ROM memory available for ROMed applets*.

5.1 Philips P8WE6017

48 kB ROM → 16kB free for ROM'd applets in Custom Mask Process

16 kB EEPROM

1300 Bytes RAM

Triple-DES coprocessor

5.2 Philips P8WE6033

96 kB ROM → 64kB free for ROM'd applets in Custom Mask Process

32 kB EEPROM

1300 Bytes RAM

Triple-DES coprocessor

* Custom applets may be submitted to Philips for inclusion into a ROM mask. The maximum package size is 16kB. The Custom Mask process is only available starting with JCOP10v2.

6 Performance figures

In the absence of standard performance tests, typical applet's operations are timed. The protocol used is T=1 at 9600 bit/sec. The reader clocks the chip at 3.71 MHz. The applets are the Visa approved versions after having been initialized and populated with keys as required for Visa VTF testing. To avoid measuring communications overhead, timing is measured between the last APDU byte sent to the reader and the first byte returned from the card.

Operation	ETUs	msec
SELECT CardManager*	162	15.1
INIT UPDATE CardManager	538	50.0
EXTERNAL AUTH CardManager	327	30.4
Install VisaCash	8025	746.3
SELECT VisaCash*	162	15.1
Initialize LOAD for VisaCash	833	77.5
Perform LOAD for VisaCash	1587	147.6
Initialize PURCHASE for VisaCash	222	20.6
Perform PURCHASE for VisaCash	1629	151.5
ReadBalance from VisaCash	96	8.9
SELECT VSDC*	582	54.1
GenerateAC (ARQC) from VSDC	2459	228.7

*: Not first SELECT to eliminate potential applet setup effects

A Revision History

1.0 Initial document

1.1 Correction of typos

1.2 AIDs of ROMed applets added

1.3 References to external specifications added

1.4 Incorrect reference to PseudoRandom constant deleted (5.1.2)

1.5 Document reformatted

1.6 Timing explanations added (6)

2.0 Specification update to cover JCOP10v2.0 (1); correction of RAM heap size (3.2.1)

2.1 Mask/Free ROM applet size clarified (5); Added comment on clock-stop feature (2.1)

B References

[1] Sun Microsystems: JavaCard 2.1.1 http://java.sun.com/products/javacard

[2] Global Platform Consortium: OpenPlatform 2.0.1' http://www.globalplatform.org/