# sdi_emv.h

Go to the documentation of this file.

```
1Â // -*- Mode: C++ -*- 2Â #ifndef SDI_EMV_H_20200406 3Â #define SDI_EMV_H_20200406 4Â  6Â #include "tlv.h"
7Â #include "sdi_enum.h" 8Â #include "sdi_emvbits.h" 9Â #include "array.h" 10Â #include "bcd.h" 11Â #include <
utility> // for std::pair 12Â  13Â #if defined _WIN32 && defined VFI_SDICLIENT_EXPORT 14Â # define DllSpec
__declspec(dllexport) 15Â #elif defined __GNUC__ && defined VFI_SDICLIENT_EXPORT 16Â # define DllSpec
__attribute__((visibility ("default"))) 17Â #else 18Â # define DllSpec 19Â #endif 20Â  21Â namespace vfisdi {
22Â  24Â typedef BitMask<40,EMVAppFlowCapabilityBits> EMVAppFlowCapabilities; 25Â  27Â typedef BitMask<
40,EMVKernelAppFlowCapabilityBits> EMVKernelAppFlowCapabilities; 28Â  30Â typedef BitMask<80,EMVFlowOptionBits>
EMVFlowOptions; 31Â  33Â typedef BitMask<40,EMVChecksumOptionBits> EMVChecksumOptions; 34Â  36Â typedef BitMask<
40,EMVTransactionOptionBits> EMVTransactionOptions; 37Â  39Â typedef BitMask<24,EMVTransactionStepBits>
EMVTransactionSteps; 40Â  41Â typedef BitMask<24,EMVDomesticOptionBits> EMVDomesticOptions; 42Â  43Â  45Â class
EMVSpecialTransactions: public Array<8> { 46Â  public: 48Â  EMVSpecialTransactions():Array<8>() {} 51Â
EMVSpecialTransactions(const unsigned char val[8]) { set(val); } 52Â  55Â  void setReversal(EMVSpecialReversal
r) { v[0]=(v[0]&0xf)|(unsigned char)(r<<4); } 58Â  EMVSpecialReversal getReversal() const { return
EMVSpecialReversal(v[0]>>4 & 0xf); } 59Â  62Â  void setRefund(EMVSpecialRefund r) { v[0]=(v[0]&0xf0)|(unsigned
char)r; } 65Â  EMVSpecialRefund getRefund() const { return EMVSpecialRefund(v[0] & 0xf); } 66Â  69Â  void
setReservation(EMVSpecialReservation r) { v[1]=(v[1]&0xf)|(unsigned char)(r<<4); } 72Â  EMVSpecialReservation
getReservation() const { return EMVSpecialReservation(v[1]>>4 & 0xf); } 73Â  76Â  void setTip(EMVSpecialTip r) {
v[1]=(v[1]&0xf0)|(unsigned char)r; } 79Â  EMVSpecialTip getTip() const { return EMVSpecialTip(v[1] & 0xf); }
80Â  83Â  void setReferral(EMVSpecialReferral r) { v[2]=(v[2]&0xf)|(unsigned char)(r<<4); } 86Â
EMVSpecialReferral getReferral() const { return EMVSpecialReferral(v[2]>>4 & 0xf); } 87Â  90Â  void
setVoice(EMVSpecialVoice r) { v[2]=(v[2]&0xf0)|(unsigned char)r; } 93Â  EMVSpecialVoice getVoice() const {
return EMVSpecialVoice(v[2] & 0xf); } 94Â  97Â  void setFallback(EMVSpecialFallback r) {
v[3]=(v[3]&0xf0)|(unsigned char)r; } 100Â  EMVSpecialFallback getFallback() const { return
EMVSpecialFallback(v[3] & 0xf); } 101Â  104Â  void setIgnoreCard(EMVSpecialIgnoreCard r) {
v[4]=(v[4]&0xf)|(unsigned char)(r<<4); } 107Â  EMVSpecialIgnoreCard getIgnoreCard() const { return
EMVSpecialIgnoreCard(v[4]>>4 & 0xf); } 108Â }; 109Â  113Â  116Â struct DllSpec EMVConf: public TLV { 117Â
public: 119Â  typedef std::vector<unsigned char> bytevector; 120Â  121Â  protected: 122Â  bool exists(unsigned
tag1) const; 123Â  bool exists(unsigned tag1, unsigned tag2) const; 124Â  125Â  public: 126Â  128Â  // primitive
data elements // 130Â  133Â  bool setIssuerID(const Array<4> &id); 136Â  bool getIssuerID(Array<4> &id) const;
137Â  140Â  bool setAID(const bytevector &aid); 142Â  template<unsigned N> bool setAID(const unsigned char
(&v)[N]) 143Â  { return setAID(bytevector(v,v+N)); } 146Â  bool getAID(bytevector &aid) const; 147Â  150Â  bool
setPOIInfo(const bytevector &poi); 153Â  bool getPOIInfo(bytevector &poi) const; 154Â  157Â  bool
setDefaultAppName(const std::string &appname); 160Â  bool getDefaultAppName(std::string &appname) const; 161Â
164Â  bool setTransactionCurrency(Currency currency); 167Â  bool getTransactionCurrency(Currency &currency)
const; 168Â  171Â  bool setTransactionCurrency(int currency); 174Â  bool getTransactionCurrency(int &currency)
const; 175Â  178Â  bool setTransactionCurrencyExp(unsigned char exp); 181Â  bool
getTransactionCurrencyExp(unsigned char &exp) const; 182Â  185Â  bool setAcquirerID(const Array<6> &acq); 188Â
bool getAcquirerID(Array<6> &acq) const; 189Â  192Â  bool setChipAppVersionNumber(const std::vector<unsigned
short> &version); 194Â  template<unsigned N> bool setChipAppVersionNumber(const unsigned short (&v)[N]) 195Â  {
return setChipAppVersionNumber(std::vector<unsigned short>(v,v+N)); } 198Â  bool
getChipAppVersionNumber(std::vector<unsigned short> &version) const; 199Â  200Â  203Â  bool
setMerchantCategory(const Array<2> &category); 206Â  bool getMerchantCategory(Array<2> &category) const; 207Â
210Â  bool setMerchantID(const std::string &mid); 213Â  bool getMerchantID(std::string &mid) const; 214Â  217Â
bool setTerminalCountryCode(unsigned short country); 220Â  bool getTerminalCountryCode(unsigned short &country)
const; 221Â  224Â  bool setFloorLimit(const BCD<6> &limit); 227Â  bool setFloorLimit(int64_t limit); 230Â  bool
getFloorLimit(BCD<6> &limit) const; 233Â  bool getFloorLimit(int64_t &limit) const; 234Â  237Â  bool
setTerminalID(const Array<8> &id); 240Â  bool getTerminalID(Array<8> &id) const; 241Â  244Â  bool
setTerminalRiskManagement(const Array<8> &trm); 247Â  bool getTerminalRiskManagement(Array<8> &trm) const; 248Â
251Â  bool setIFDSerial(const Array<8> &serial); 254Â  bool getIFDSerial(Array<8> &serial) const; 255Â  258Â
bool setTerminalCapabilities(const Array<3> &caps); 261Â  bool getTerminalCapabilities(Array<3> &caps) const;
262Â  265Â  bool setTerminalType(unsigned char type); 268Â  bool getTerminalType(unsigned char &type) const;
269Â  272Â  bool setPOSEntryMode(unsigned char mode); 275Â  bool getPOSEntryMode(unsigned char &mode) const;
276Â  279Â  bool setAdditionalCapabilities(const Array<5> &caps); 282Â  bool getAdditionalCapabilities(Array<5>
&caps) const; 283Â  286Â  bool setMerchantNameLocation(const std::string &nameloc); 289Â  bool
getMerchantNameLocation(std::string &nameloc) const; 290Â  293Â  bool setTransactionCategory(unsigned char cat);
296Â  bool getTransactionCategory(unsigned char &cat) const; 297Â  300Â  bool setInterchangeProfile(const Array<
3> &icc); 303Â  bool getInterchangeProfile(Array<3> &icc) const; 304Â  307Â  bool setMerchantType(unsigned char
type); 310Â  bool getMerchantType(unsigned char &type) const; 311Â  314Â  bool
setTerminalTransactionInformation(const Array<3> &info); 317Â  bool getTerminalTransactionInformation(Array<3>
&info) const; 318Â  321Â  bool setAppProgramID(const bytevector &id); 323Â  template<unsigned N> bool
setAppProgramID(const unsigned char (&v)[N]) 324Â  { return setAppProgramID(bytevector(v,v+N)); } 327Â  bool
getAppProgramID(bytevector &id) const; 328Â  331Â  bool setDataStoreRequestedOperatorID(const Array<8> &id);
334Â  bool getDataStoreRequestedOperatorID(Array<8> &id) const; 335Â  338Â  bool setReceiptRequiredLimit(const
BCD<6> &limit); 341Â  bool setReceiptRequiredLimit(int64_t limit); 344Â  bool getReceiptRequiredLimit(BCD<6>
&limit) const; 347Â  bool getReceiptRequiredLimit(int64_t &limit) const; 348Â  351Â  bool setOptionStatus(const
Array<2> &status); 354Â  bool getOptionStatus(Array<2> &status) const; 355Â  358Â  bool
setTransactionQualifier(const Array<4> &q); 361Â  bool getTransactionQualifier(Array<4> &q) const; 362Â  365Â
```

```
bool setMSRVersionNumber(const std::vector<unsigned short> &ver); 367Â  template<unsigned N> bool
setMSRVersionNumber(const unsigned short (&v)[N]) 368Â  { return setMSRVersionNumber(std::vector<unsigned short>
(v,v+N)); } 371Â  bool getMSRVersionNumber(std::vector<unsigned short> &ver) const; 372Â  375Â  bool
setReaderCapabilities(unsigned char caps); 378Â  bool getReaderCapabilities(unsigned char &caps) const; 379Â
382Â  bool setTransactionCapabilities(const Array<4> &caps); 385Â  bool getTransactionCapabilities(Array<4>
&caps) const; 386Â  389Â  bool setTerminalTransactionData(const bytevector &data); 391Â  template<unsigned N>
bool setTerminalTransactionData(const unsigned char (&v)[N]) 392Â  { return
setTerminalTransactionData(bytevector(v,v+N)); } 395Â  bool getTerminalTransactionData(bytevector &data) const;
396Â  399Â  bool setMerchantCustomData(const Array<20> &data); 402Â  bool getMerchantCustomData(Array<20> &data)
const; 403Â  406Â  bool getLibVersion(std::string &version) const; 407Â  410Â  bool getEMVCOChecksum(std::string
&chk) const; 411Â  414Â  bool setEMVCOChecksumParameter(const Array<5> &cp); 417Â  bool
setEMVCOChecksumParameter(const EMVChecksumOptions &cp) 418Â  { return setEMVCOChecksumParameter(cp.array()); }
421Â  bool getEMVCOChecksumParameter(Array<5> &cp) const; 424Â  bool
getEMVCOChecksumParameter(EMVChecksumOptions &cp) const 425Â  { return getEMVCOChecksumParameter(cp.array()); }
426Â  430Â  bool setOfflineOnly(bool flag); 434Â  bool getOfflineOnly(bool &flag) const; 435Â  439Â  bool
setTACIACDefaultHandling(unsigned char def); 443Â  bool getTACIACDefaultHandling(unsigned char &def) const;
444Â  447Â  bool setFallbackMIDs(const Array<3> &mid); 450Â  bool getFallbackMIDs(Array<3> &mid) const; 451Â
454Â  bool setEMVFallback(unsigned char fb); 457Â  bool getEMVFallback(unsigned char &fb) const; 458Â  461Â
bool setSpecialTransactions(const EMVSpecialTransactions &tx); 464Â  bool
getSpecialTransactions(EMVSpecialTransactions &tx) const; 465Â  470Â  bool setPriorityApplications(const
std::vector<bytevector> &apps); 473Â  bool getPriorityApplications(std::vector<bytevector> &apps) const; 474Â
477Â  bool setASI(unsigned char asi); 480Â  bool getASI(unsigned char &asi) const; 481Â  484Â  bool
setTACDenial(const Array<5> &tac); 487Â  bool getTACDenial(Array<5> &tac) const; 488Â  491Â  bool
setTACOnline(const Array<5> &tac); 494Â  bool getTACOnline(Array<5> &tac) const; 495Â  498Â  bool
setTACDefault(const Array<5> &tac); 501Â  bool getTACDefault(Array<5> &tac) const; 502Â  505Â  bool
setThreshold(unsigned th); 508Â  bool getThreshold(unsigned &th) const; 509Â  512Â  bool
setMaxPercentOnline(const BCD<1> &p); 515Â  bool setMaxPercentOnline(int p); 518Â  bool getMaxPercentOnline(BCD<
1> &p) const; 521Â  bool getMaxPercentOnline(int &p) const; 522Â  525Â  bool setPercentOnline(const BCD<1> &p);
528Â  bool setPercentOnline(int p); 531Â  bool getPercentOnline(BCD<1> &p) const; 534Â  bool
getPercentOnline(int &p) const; 535Â  538Â  bool setDefaultTDOL(const bytevector &tdol); 540Â  template<unsigned
N> bool setDefaultTDOL(const unsigned char (&v)[N]) 541Â  { return setDefaultTDOL(bytevector(v,v+N)); } 544Â
bool getDefaultTDOL(bytevector &tdol) const; 545Â  548Â  bool setDefaultDDOL(const bytevector &ddol); 550Â
template<unsigned N> bool setDefaultDDOL(const unsigned char (&v)[N]) 551Â  { return
setDefaultDDOL(bytevector(v,v+N)); } 554Â  bool getDefaultDDOL(bytevector &ddol) const; 555Â  558Â  bool
setAdditionalTags(const bytevector &tags); 560Â  template<unsigned N> bool setAdditionalTags(const unsigned char
(&v)[N]) 561Â  { return setAdditionalTags(bytevector(v,v+N)); } 564Â  bool getAdditionalTags(bytevector &tags)
const; 565Â  568Â  bool setMandatoryTags(const std::vector<unsigned> &tags); 570Â  template<unsigned N> bool
setMandatoryTags(const unsigned (&v)[N]) 571Â  { return setMandatoryTags(std::vector<unsigned>(v,v+N)); } 574Â
bool getMandatoryTags(std::vector<unsigned> &tags) const; 575Â  578Â  bool setAppFlowCapabilities(const Array<5>
&caps); 581Â  bool setAppFlowCapabilities(const EMVAppFlowCapabilities &caps) 582Â  { return
setAppFlowCapabilities(caps.array()); } 585Â  bool getAppFlowCapabilities(Array<5> &caps) const; 588Â  bool
getAppFlowCapabilities(EMVAppFlowCapabilities &caps) const 589Â  { return getAppFlowCapabilities(caps.array());
} 590Â  593Â  bool setAdditionalTagsCRD(const bytevector &tags); 595Â  template<unsigned N> bool
setAdditionalTagsCRD(const unsigned char (&v)[N]) 596Â  { return setAdditionalTagsCRD(bytevector(v,v+N)); }
599Â  bool getAdditionalTagsCRD(bytevector &tags) const; 600Â  603Â  bool setEMVApplication(unsigned char emv);
606Â  bool getEMVApplication(unsigned char &emv) const; 607Â  610Â  bool setAIPNoCVM(unsigned char cvm); 613Â
bool getAIPNoCVM(unsigned char &cvm) const; 614Â  617Â  bool setCustomCVM(unsigned char cvm); 620Â  bool
getCustomCVM(unsigned char &cvm) const; 621Â  624Â  bool setAdditionalTerminalCapabilities(const Array<5>
&caps); 627Â  bool getAdditionalTerminalCapabilities(Array<5> &caps) const; 628Â  631Â  bool
setCDAProcessing(unsigned char cda); 634Â  bool getCDAProcessing(unsigned char &cda) const; 635Â  638Â  bool
getObfuscatedPANString(std::string &pan) const; 639Â  642Â  bool getObfuscatedTrack2String(std::string &track)
const; 643Â  646Â  bool setSecurityLimit(unsigned limit); 649Â  bool getSecurityLimit(unsigned &limit) const;
650Â  653Â  bool setCapabilitiesBelowLimit(const Array<3> &caps); 656Â  bool getCapabilitiesBelowLimit(Array<3>
&caps) const; 657Â  660Â  bool setBeepVolume(unsigned short volume); 663Â  bool getBeepVolume(unsigned short
&volume) const; 664Â  667Â  bool getKernelVersion(std::string &version) const; 668Â  671Â  bool
setConfigCommand(unsigned char cmd); 674Â  bool getConfigCommand(unsigned char &cmd) const; 675Â  678Â  bool
setSupportedLanguages(const bytevector &lang); 680Â  template<unsigned N> bool setSupportedLanguages(const
unsigned char (&v)[N]) 681Â  { return setSupportedLanguages(bytevector(v,v+N)); } 684Â  bool
getSupportedLanguages(bytevector &lang) const; 685Â  688Â  bool setRetryLimit(unsigned char limit); 691Â  bool
getRetryLimit(unsigned char &limit) const; 692Â  698Â  bool getL1DriverVersion(std::string &version) const;
699Â  703Â  bool getCtL1DriverVersion(std::string &version) const; 704Â  708Â  bool
getCtlsL1DriverVersion(std::string &version) const; 709Â  712Â  bool setShortKernelID(unsigned char id); 715Â
bool getShortKernelID(unsigned char &id) const; 716Â  719Â  bool setProceedToFirstWriteFlag(unsigned char flag);
722Â  bool getProceedToFirstWriteFlag(unsigned char &flag) const; 723Â  726Â  bool setTagsToRead(const
bytevector &ttr); 728Â  template<unsigned N> bool setTagsToRead(const unsigned char (&v)[N]) 729Â  { return
setTagsToRead(bytevector(v,v+N)); } 732Â  bool getTagsToRead(bytevector &ttr) const; 733Â  736Â  bool
setCardDataInputCapability(unsigned char cap); 739Â  bool getCardDataInputCapability(unsigned char &cap) const;
740Â  743Â  bool setChipCVMAboveLimit(unsigned char cvm); 746Â  bool getChipCVMAboveLimit(unsigned char &cvm)
```

```
const; 747Â  750Â  bool setChipCVMBelowLimit(unsigned char cvm); 753Â  bool getChipCVMBelowLimit(unsigned char
&cvm) const; 754Â  757Â  bool setKernelConfiguration(unsigned char conf); 760Â  bool
getKernelConfiguration(unsigned char &conf) const; 761Â  764Â  bool setTornTransactionLifetime(unsigned short
ttl); 767Â  bool getTornTransactionLifetime(unsigned short &ttl) const; 768Â  771Â  bool
setTornTransactionNumber(unsigned char num); 774Â  bool getTornTransactionNumber(unsigned char &num) const;
775Â  778Â  bool setMagstripeCVMAboveLimit(unsigned char cvm); 781Â  bool getMagstripeCVMAboveLimit(unsigned
char &cvm) const; 782Â  785Â  bool setSecurityCapability(unsigned char cap); 788Â  bool
getSecurityCapability(unsigned char &cap) const; 789Â  792Â  bool setTransactionLimitNoCVMOnDevice(const BCD<6>
&limit); 795Â  bool setTransactionLimitNoCVMOnDevice(int64_t limit); 798Â  bool
getTransactionLimitNoCVMOnDevice(BCD<6> &limit) const; 801Â  bool getTransactionLimitNoCVMOnDevice(int64_t
&limit) const; 802Â  805Â  bool setTransactionLimitCVMOnDevice(const BCD<6> &limit); 808Â  bool
setTransactionLimitCVMOnDevice(int64_t limit); 811Â  bool getTransactionLimitCVMOnDevice(BCD<6> &limit) const;
814Â  bool getTransactionLimitCVMOnDevice(int64_t &limit) const; 815Â  818Â  bool
setDataExchangeTimeout(unsigned short tout); 821Â  bool getDataExchangeTimeout(unsigned short &tout) const;
822Â  825Â  bool setMagstripeCVMBelowLimit(unsigned char cvm); 828Â  bool getMagstripeCVMBelowLimit(unsigned
char &cvm) const; 829Â  832Â  bool setMessageHoldTime(const BCD<3> &time); 835Â  bool setMessageHoldTime(int
time); 838Â  bool getMessageHoldTime(BCD<3> &time) const; 841Â  bool getMessageHoldTime(int &time) const; 842Â
845Â  bool setFieldOffTime(unsigned char time); 848Â  bool getFieldOffTime(unsigned char &time) const; 849Â
852Â  bool setPhoneMessageTable(const bytevector &pmt); 854Â  template<unsigned N> bool
setPhoneMessageTable(const unsigned char (&v)[N]) 855Â  { return setPhoneMessageTable(bytevector(v,v+N)); }
858Â  bool getPhoneMessageTable(bytevector &pmt) const; 859Â  862Â  bool
setRelayResistanceMinGracePeriod(unsigned short time); 865Â  bool getRelayResistanceMinGracePeriod(unsigned
short &time) const; 866Â  869Â  bool setRelayResistanceMaxGracePeriod(unsigned short time); 872Â  bool
getRelayResistanceMaxGracePeriod(unsigned short &time) const; 873Â  876Â  bool
setRelayResistanceCAPDUTime(unsigned short time); 879Â  bool getRelayResistanceCAPDUTime(unsigned short &time)
const; 880Â  883Â  bool setRelayResistanceRAPDUTime(unsigned short time); 886Â  bool
getRelayResistanceRAPDUTime(unsigned short &time) const; 887Â  890Â  bool setRelayResistanceAccuracy(unsigned
short time); 893Â  bool getRelayResistanceAccuracy(unsigned short &time) const; 894Â  897Â  bool
setRelayResistanceTimeMismatchThreshold(unsigned char percent); 900Â  bool
getRelayResistanceTimeMismatchThreshold(unsigned char &percent) const; 901Â  904Â  bool setFlowOptions(const
Array<10> &opt); 907Â  bool setFlowOptions(const EMVFlowOptions &opt) 908Â  { return
setFlowOptions(opt.array()); } 911Â  bool getFlowOptions(Array<10> &opt) const; 914Â  bool
getFlowOptions(EMVFlowOptions &opt) const 915Â  { return getFlowOptions(opt.array()); } 916Â  919Â  bool
setCtlsTerminalLimit(unsigned limit); 922Â  bool getCtlsTerminalLimit(unsigned &limit) const; 923Â  926Â  bool
setKernelID(const Array<3> &id); 929Â  bool setKernelID(unsigned id); 932Â  bool getKernelID(Array<3> &id)
const; 935Â  bool getKernelID(unsigned &id) const; 936Â  939Â  bool getEntryPointChecksum(std::string &chk)
const; 940Â  943Â  bool getKernelChecksum(std::string &chk) const; 944Â  947Â  bool setRetapFieldOff(unsigned
char offtime); 950Â  bool getRetapFieldOff(unsigned char &offtime) const; 951Â  954Â  bool
setInternalKernelID(const Array<3> &id); 957Â  bool setInternalKernelID(unsigned id); 960Â  bool
getInternalKernelID(Array<3> &id) const; 963Â  bool getInternalKernelID(unsigned &id) const; 964Â  967Â  bool
setTECSupport(unsigned char ts); 970Â  bool getTECSupport(unsigned char &ts) const; 971Â  974Â  bool
setKernelAppFlowCapabilities(const Array<5> &caps); 977Â  bool setKernelAppFlowCapabilities(const
EMVKernelAppFlowCapabilities &caps) 978Â  { return setKernelAppFlowCapabilities(caps.array()); } 981Â  bool
getKernelAppFlowCapabilities(Array<5> &caps) const; 984Â  bool
getKernelAppFlowCapabilities(EMVKernelAppFlowCapabilities &caps) const 985Â  { return
getKernelAppFlowCapabilities(caps.array()); } 986Â  989Â  bool setCtlsTransactionLimit(const BCD<6> &limit);
992Â  bool setCtlsTransactionLimit(int64_t limit); 995Â  bool getCtlsTransactionLimit(BCD<6> &limit) const;
998Â  bool getCtlsTransactionLimit(int64_t &limit) const; 999Â  1002Â  bool setCVMRequiredLimit(const BCD<6>
&limit); 1005Â  bool setCVMRequiredLimit(int64_t limit); 1008Â  bool getCVMRequiredLimit(BCD<6> &limit) const;
1011Â  bool getCVMRequiredLimit(int64_t &limit) const; 1012Â  1015Â  bool setRiskManagementThreshold(const BCD<6
> &t); 1018Â  bool setRiskManagementThreshold(int64_t t); 1021Â  bool getRiskManagementThreshold(BCD<6> &t)
const; 1024Â  bool getRiskManagementThreshold(int64_t &t) const; 1025Â  1028Â  bool
setRiskManagementTargetPercentage(const BCD<1> &p); 1031Â  bool setRiskManagementTargetPercentage(int p); 1034Â
bool getRiskManagementTargetPercentage(BCD<1> &p) const; 1037Â  bool getRiskManagementTargetPercentage(int &p)
const; 1038Â  1041Â  bool setRiskManagementMaxTargetPercentage(const BCD<1> &p); 1044Â  bool
setRiskManagementMaxTargetPercentage(int p); 1047Â  bool getRiskManagementMaxTargetPercentage(BCD<1> &p) const;
1050Â  bool getRiskManagementMaxTargetPercentage(int &p) const; 1051Â  1054Â  bool
setDynamicReaderLimitsOptions(unsigned char opts); 1057Â  bool getDynamicReaderLimitsOptions(unsigned char
&opts) const; 1058Â  1061Â  bool setTransactionLimitCash(const BCD<6> &limit); 1064Â  bool
setTransactionLimitCash(int64_t limit); 1067Â  bool getTransactionLimitCash(BCD<6> &limit) const; 1070Â  bool
getTransactionLimitCash(int64_t &limit) const; 1071Â  1074Â  bool setCombinationOptions(const Array<2> &opt);
1077Â  bool getCombinationOptions(Array<2> &opt) const; 1078Â  1079Â  1082Â  bool setRemovalTimeout(unsigned
short tout); 1085Â  bool getRemovalTimeout(unsigned short &tout) const; 1086Â  1089Â  bool
setCallbackTimeout(unsigned short tout); 1092Â  bool getCallbackTimeout(unsigned short &tout) const; 1093Â
1096Â  bool setTornTransactionInterval(const Array<2> &interval); 1099Â  bool getTornTransactionInterval(Array<2
> &interval) const; 1100Â  1103Â  bool setCtlsKernelCapabilities(const Array<5> &caps); 1106Â  bool
getCtlsKernelCapabilities(Array<5> &caps) const; 1107Â  1110Â  bool setMTOL(const bytevector &mtol); 1112Â
template<unsigned N> bool setMTOL(const unsigned char (&v)[N]) 1113Â  { return setMTOL(bytevector(v,v+N)); }
```

```
1116Â  bool getMTOL(bytevector &mtol) const; 1117Â  1120Â  bool setUnpredictableNumberRange(unsigned char
range); 1123Â  bool getUnpredictableNumberRange(unsigned char &range) const; 1124Â  1127Â  bool
setTermcapAboveCVMLimit(const Array<3> &tcap); 1130Â  bool getTermcapAboveCVMLimit(Array<3> &tcap) const; 1131Â
1134Â  bool setTermcapBelowCVMLimit(const Array<3> &tcap); 1137Â  bool getTermcapBelowCVMLimit(Array<3> &tcap)
const; 1138Â  1141Â  bool setTACSwitch(const Array<5> &sw); 1144Â  bool getTACSwitch(Array<5> &sw) const; 1145Â
1148Â  bool setIACSwitch(const Array<5> &sw); 1151Â  bool getIACSwitch(Array<5> &sw) const; 1152Â  1155Â  bool
setDataExchangeTags(const bytevector &tags); 1157Â  template<unsigned N> bool setDataExchangeTags(const unsigned
char (&v)[N]) 1158Â  { return setDataExchangeTags(bytevector(v,v+N)); } 1161Â  bool
getDataExchangeTags(bytevector &tags) const; 1162Â  1165Â  bool setDCRListIDs(const bytevector &idlist); 1168Â
bool getDCRListIDs(bytevector &idlist) const; 1169Â  1172Â  bool setDCRListRanges(const bytevector &rangelist);
1175Â  bool getDCRListRanges(bytevector &rangelist) const; 1176Â  1179Â  bool
setApplicationAuthTransactionType(unsigned char type); 1182Â  bool getApplicationAuthTransactionType(unsigned
char &type) const; 1183Â  1186Â  bool setsetCtlsTransactionLimitODCVM(const BCD<6> &limit); 1189Â  bool
setsetCtlsTransactionLimitODCVM(int64_t limit); 1192Â  bool getsetCtlsTransactionLimitODCVM(BCD<6> &limit)
const; 1195Â  bool getsetCtlsTransactionLimitODCVM(int64_t &limit) const; 1196Â  1199Â  bool
setTransactionTypesSale(const Array<3> &type); 1202Â  bool getTransactionTypesSale(Array<3> &type) const; 1203Â
1206Â  bool setTransactionTypesCash(const Array<3> &type); 1209Â  bool getTransactionTypesCash(Array<3> &type)
const; 1210Â  1213Â  bool setTransactionTypesCashback(const Array<3> &type); 1216Â  bool
getTransactionTypesCashback(Array<3> &type) const; 1217Â  1220Â  bool setBeepFreqSuccess(unsigned short freq);
1223Â  bool getBeepFreqSuccess(unsigned short &freq) const; 1224Â  1227Â  bool setTransactionTypesRefund(const
Array<3> &type); 1230Â  bool getTransactionTypesRefund(Array<3> &type) const; 1231Â  1234Â  bool
setBeepFreqAlert(unsigned short freq); 1237Â  bool getBeepFreqAlert(unsigned short &freq) const; 1238Â  1241Â
bool setSecondTapDelay(unsigned char delay); 1244Â  bool getSecondTapDelay(unsigned char &delay) const; 1245Â
1248Â  bool setHostTimeout(unsigned char tout); 1251Â  bool getHostTimeout(unsigned char &tout) const; 1252Â
1253Â  1255Â  // constructed data elements // 1257Â  1260Â  bool setHotlist(const bytevector &hotlist); 1262Â
template<unsigned N> bool setHotlist(const unsigned char (&v)[N]) 1263Â  { return setHotlist(bytevector(v,v+N));
} 1266Â  bool setHotlist(const TLV &hotlist); 1269Â  bool getHotlist(bytevector &hotlist) const; 1272Â  bool
getHotlist(TLV &hotlist) const; 1273Â  1276Â  bool setTagsToWriteBeforeGenAC(const TLV &ttw); 1279Â  bool
getTagsToWriteBeforeGenAC(TLV &ttw) const; 1280Â  1283Â  bool setTagsToWriteAfterGenAC(const TLV &ttw); 1286Â
bool getTagsToWriteAfterGenAC(TLV &ttw) const; 1287Â  1289Â  struct DynamicReaderLimits { 1290Â  bytevector
appProgramID; 1291Â  BCD<6> floorLimit; 1292Â  BCD<6> transactionLimit; 1293Â  BCD<6> CVMRequiredLimit; 1294Â
unsigned char options; 1295Â #if __cplusplus >= 201103 1296Â  DynamicReaderLimits(const bytevector &aid_, 1297Â
const BCD<6> &floorLimit_, 1298Â  const BCD<6> &transactionLimit_, 1299Â  const BCD<6> &CVMRequiredLimit_,
1300Â  unsigned char options_) 1301Â  : appProgramID(aid_), 1302Â  floorLimit(floorLimit_), 1303Â
transactionLimit(transactionLimit_), 1304Â  CVMRequiredLimit(CVMRequiredLimit_), 1305Â  options(options_) {}
1306Â  DynamicReaderLimits() = default; 1307Â #endif 1308Â  }; 1309Â  1312Â  bool setDynamicReaderLimits(const
std::vector<DynamicReaderLimits> &limits); 1315Â  bool getDynamicReaderLimits(std::vector<DynamicReaderLimits>
&limits) const; 1316Â  1318Â  // issuer specific 1319Â  1322Â  bool setTransactionTypeIK(unsigned char type);
1325Â  bool getTransactionTypeIK(unsigned char &type) const; 1326Â  1329Â  bool setFloorLimitIK(const BCD<6>
&limit); 1332Â  bool setFloorLimitIK(int64_t limit); 1335Â  bool getFloorLimitIK(BCD<6> &limit) const; 1338Â
bool getFloorLimitIK(int64_t &limit) const; 1339Â  1342Â  bool setCVMRequiredLimitPK(const BCD<6> &limit);
1345Â  bool setCVMRequiredLimitPK(int64_t limit); 1348Â  bool getCVMRequiredLimitPK(BCD<6> &limit) const; 1351Â
bool getCVMRequiredLimitPK(int64_t &limit) const; 1352Â  1355Â  bool setFloorlimitPK(const BCD<6> &limit);
1358Â  bool setFloorlimitPK(int64_t limit); 1361Â  bool getFloorlimitPK(BCD<6> &limit) const; 1364Â  bool
getFloorlimitPK(int64_t &limit) const; 1365Â  1368Â  bool setFloorLimitMIR(const BCD<6> &limit); 1371Â  bool
setFloorLimitMIR(int64_t limit); 1374Â  bool getFloorLimitMIR(BCD<6> &limit) const; 1377Â  bool
getFloorLimitMIR(int64_t &limit) const; 1378Â  1381Â  bool setTransactionLimitNoCVMMIR(const BCD<6> &limit);
1384Â  bool setTransactionLimitNoCVMMIR(int64_t limit); 1387Â  bool getTransactionLimitNoCVMMIR(BCD<6> &limit)
const; 1390Â  bool getTransactionLimitNoCVMMIR(int64_t &limit) const; 1391Â  1394Â  bool
setTransactionLimitNoCDCVMMIR(const BCD<6> &limit); 1397Â  bool setTransactionLimitNoCDCVMMIR(int64_t limit);
1400Â  bool getTransactionLimitNoCDCVMMIR(BCD<6> &limit) const; 1403Â  bool
getTransactionLimitNoCDCVMMIR(int64_t &limit) const; 1404Â  1407Â  bool setTransactionLimitCDCVMMIR(const BCD<6>
&limit); 1410Â  bool setTransactionLimitCDCVMMIR(int64_t limit); 1413Â  bool getTransactionLimitCDCVMMIR(BCD<6>
&limit) const; 1416Â  bool getTransactionLimitCDCVMMIR(int64_t &limit) const; 1417Â  1420Â  bool
setTPMCapabilitiesMIR(const Array<2> &caps); 1423Â  bool getTPMCapabilitiesMIR(Array<2> &caps) const; 1424Â
1427Â  bool setTransactionRecoveryLimitMIR(unsigned char limit); 1430Â  bool
getTransactionRecoveryLimitMIR(unsigned char &limit) const; 1431Â  1434Â  bool setAdditionalVersionNumbers(const
bytevector &ver); 1436Â  template<unsigned N> bool setAdditionalVersionNumbers(const unsigned char (&v)[N])
1437Â  { return setAdditionalVersionNumbers(bytevector(v,v+N)); } 1440Â  bool
getAdditionalVersionNumbers(bytevector &ver) const; 1441Â  1442Â  1444Â  1447Â  bool setTACDefaultMK(const Array
<5> &tac); 1450Â  bool getTACDefaultMK(Array<5> &tac) const; 1451Â  1454Â  bool setTACDenialMK(const Array<5>
&tac); 1457Â  bool getTACDenialMK(Array<5> &tac) const; 1458Â  1461Â  bool setTACOnlineMK(const Array<5> &tac);
1464Â  bool getTACOnlineMK(Array<5> &tac) const; 1465Â  1468Â  bool setFloorLimitMK(const BCD<6> &limit); 1471Â
bool setFloorLimitMK(int64_t limit); 1474Â  bool getFloorLimitMK(BCD<6> &limit) const; 1477Â  bool
getFloorLimitMK(int64_t &limit) const; 1478Â  1481Â  bool setCVMRequiredLimitMK(const BCD<6> &limit); 1484Â
bool setCVMRequiredLimitMK(int64_t limit); 1487Â  bool getCVMRequiredLimitMK(BCD<6> &limit) const; 1490Â  bool
getCVMRequiredLimitMK(int64_t &limit) const; 1491Â  1492Â }; 1493Â  1494Â  1500Â  1501Â  1504Â struct DllSpec
EMVTxn: public TLV { 1505Â  public: 1507Â  typedef std::vector<unsigned char> bytevector; 1508Â  1509Â
```

```
protected: 1510Â  bool exists(unsigned tag1) const; 1511Â  bool exists(unsigned tag1, unsigned tag2) const;
1512Â  1513Â  public: 1514Â  1516Â  // primitive data elements // 1518Â  1521Â  bool getIssuerID(Array<4> &id)
const; 1522Â  1525Â  bool setAID(const bytevector &aid); 1527Â  template<unsigned N> bool setAID(const unsigned
char (&v)[N]) 1528Â  { return setAID(bytevector(v,v+N)); } 1531Â  bool getAID(bytevector &aid) const; 1532Â
1535Â  bool getAppName(std::string &label) const; 1536Â  1539Â  bool getAgreedAppName(std::string &name) const;
1540Â  1543Â  bool getPAN(Array<10> &pan) const; 1544Â  1547Â  bool getAIP(Array<2> &aip) const; 1548Â  1551Â
bool getDFName(bytevector &name) const; 1552Â  1555Â  bool getApplicationPriorityID(unsigned char &id) const;
1556Â  1559Â  bool setAuthCode(const Array<6> &code); 1562Â  bool getAuthCode(Array<6> &code) const; 1563Â
1566Â  bool setAuthResponseCode(const Array<2> &auth); 1569Â  bool getAuthResponseCode(Array<2> &auth) const;
1570Â  1573Â  bool getCVMList(Array<28> &list) const; 1574Â  1577Â  bool getTVR(Array<5> &tvr) const; 1578Â
1581Â  bool setTransactionDate(const Array<3> &date); 1584Â  bool setTransactionDate(const std::string &date);
1587Â  bool getTransactionDate(Array<3> &date) const; 1590Â  bool getTransactionDate(std::string &date) const;
1591Â  1594Â  bool getTransactionStatusInfo(Array<2> &status) const; 1595Â  1598Â  bool
setTransactionType(unsigned char type); 1601Â  bool getTransactionType(unsigned char &type) const; 1602Â  1605Â
bool setCtlsLEDState(unsigned char state); 1608Â  bool getCtlsLEDState(unsigned char &state) const; 1609Â
1612Â  bool setServerPollTimeout(unsigned char tout); 1615Â  bool getServerPollTimeout(unsigned char &tout)
const; 1616Â  1619Â  bool setBeepScenario(unsigned char bs); 1622Â  bool getBeepScenario(unsigned char &bs)
const; 1623Â  1626Â  bool getCardholderName(std::string &name) const; 1627Â  1630Â  bool setExpirationDate(const
Array<3> &date); 1633Â  bool getExpirationDate(Array<3> &date) const; 1634Â  1637Â  bool getEffectiveDate(Array<
3> &date) const; 1640Â  bool getEffectiveDate(std::string &date) const; 1641Â  1644Â  bool
setIssuerCountryCode(unsigned short country); 1647Â  bool getIssuerCountryCode(unsigned short &country) const;
1648Â  1651Â  bool setTransactionCurrency(Currency currency); 1654Â  bool getTransactionCurrency(Currency
&currency) const; 1655Â  1658Â  bool setTransactionCurrency(int currency); 1661Â  bool
getTransactionCurrency(int &currency) const; 1662Â  1665Â  bool getLanguagePreference(std::string &lang) const;
1666Â  1669Â  bool getServiceCode(Array<2> &code) const; 1670Â  1673Â  bool getPANSequenceNumber(BCD<1> &num)
const; 1676Â  bool getPANSequenceNumber(int &num) const; 1677Â  1680Â  bool setTransactionCurrencyExp(unsigned
char exp); 1683Â  bool getTransactionCurrencyExp(unsigned char &exp) const; 1684Â  1687Â  bool
setAccountType(unsigned char type); 1690Â  bool getAccountType(unsigned char &type) const; 1691Â  1694Â  bool
setAmount(const BCD<6> &amount); 1697Â  bool setAmount(int64_t amount); 1700Â  bool getAmount(BCD<6> &amount)
const; 1703Â  bool getAmount(int64_t &amount) const; 1704Â  1707Â  bool setCashbackAmount(const BCD<6> &amount);
1710Â  bool setCashbackAmount(int64_t amount); 1713Â  bool getCashbackAmount(BCD<6> &amount) const; 1716Â  bool
getCashbackAmount(int64_t &amount) const; 1717Â  1720Â  bool getTerminalAID(bytevector &aid) const; 1721Â
1724Â  bool getKernelAppVersionNumber(Array<2> &version) const; 1725Â  1728Â  bool getChipAppVersionNumber(Array
<2> &version) const; 1729Â  1732Â  bool getASRPD(bytevector &asrpd) const; 1733Â  1736Â  bool
getCardholderNameExt(std::string &name) const; 1737Â  1740Â  bool getIACDefault(Array<5> &iac) const; 1741Â
1744Â  bool getIACDenial(Array<5> &iac) const; 1745Â  1748Â  bool getIACOnline(Array<5> &iac) const; 1749Â
1752Â  bool getIssuerAppData(bytevector &idata) const; 1753Â  1756Â  bool getIssuerCodeTableID(unsigned char
&id) const; 1757Â  1760Â  bool getPreferredName(std::string &label) const; 1761Â  1764Â  bool
getMerchantID(std::string &mid) const; 1765Â  1768Â  bool getTerminalCountryCode(unsigned short &country) const;
1769Â  1772Â  bool setFloorLimit(const BCD<6> &limit); 1775Â  bool setFloorLimit(int64_t limit); 1778Â  bool
getFloorLimit(BCD<6> &limit) const; 1781Â  bool getFloorLimit(int64_t &limit) const; 1782Â  1785Â  bool
getIFDSerial(Array<8> &serial) const; 1786Â  1789Â  bool setTransactionTime(const Array<3> &time); 1792Â  bool
setTransactionTime(const std::string &time); 1795Â  bool getTransactionTime(Array<3> &time) const; 1798Â  bool
getTransactionTime(std::string &time) const; 1799Â  1802Â  bool setCryptogram(const Array<8> &crypt); 1805Â
bool getCryptogram(Array<8> &crypt) const; 1806Â  1809Â  bool getCryptogramInfo(unsigned char &info) const;
1810Â  1813Â  bool getTerminalCapabilities(Array<3> &caps) const; 1814Â  1817Â  bool getCVMResults(Array<3>
&cvmr) const; 1818Â  1821Â  bool getTerminalType(unsigned char &type) const; 1822Â  1825Â  bool getATC(Array<2>
&atc) const; 1826Â  1829Â  bool setUnpredictableNumber(unsigned num); 1832Â  bool
getUnpredictableNumber(unsigned &num) const; 1833Â  1836Â  bool getPOSEntryMode(unsigned char &mode) const;
1837Â  1840Â  bool getAdditionalCapabilities(Array<5> &caps) const; 1841Â  1844Â  bool
setTransactionCounter(const BCD<4> &counter); 1847Â  bool setTransactionCounter(int counter); 1850Â  bool
getTransactionCounter(BCD<4> &counter) const; 1853Â  bool getTransactionCounter(int &counter) const; 1854Â
1857Â  bool setApplicationCurrency(Currency currency); 1860Â  bool getApplicationCurrency(Currency &currency)
const; 1861Â  1864Â  bool getDataAuthCode(Array<2> &code) const; 1865Â  1868Â  bool
getICCDynamicNumber(bytevector &dn) const; 1869Â  1872Â  bool getTransactionCategory(unsigned char &cat) const;
1873Â  1876Â  bool getAvailableOfflineAmount(BCD<6> &amount) const; 1879Â  bool
getAvailableOfflineAmount(int64_t &amount) const; 1880Â  1883Â  bool getTransactionQualifier(Array<4> &q) const;
1884Â  1887Â  bool setVisaCTQ(const Array<2> &ctq); 1890Â  bool getVisaCTQ(Array<2> &ctq) const; 1891Â  1894Â
bool setMasterAID(const bytevector &aid); 1896Â  template<unsigned N> bool setMasterAID(const unsigned char
(&v)[N]) 1897Â  { return setMasterAID(bytevector(v,v+N)); } 1900Â  bool getMasterAID(bytevector &aid) const;
1901Â  1904Â  bool setBuildAppList(unsigned char opt); 1907Â  bool getBuildAppList(unsigned char &opt) const;
1908Â  1911Â  bool getReaderNumber(unsigned char &num) const; 1912Â  1915Â  bool
getNonCriticalScriptResults(std::vector<Array<5> > &results) const; 1916Â  1919Â  bool
getCriticalScriptResults(std::vector<Array<5> > &results) const; 1920Â  1923Â  bool setFallbackMIDs(const Array<
3> &mid); 1926Â  bool getFallbackMIDs(Array<3> &mid) const; 1927Â  1930Â  bool setEMVFallback(unsigned char fb);
1933Â  bool getEMVFallback(unsigned char &fb) const; 1934Â  1937Â  bool getObfuscatedPAN(bytevector &pan) const;
1938Â  1941Â  bool getObfuscatedTrack2(bytevector &track) const; 1942Â  1945Â  bool setTACDenial(const Array<5>
&tac); 1948Â  bool getTACDenial(Array<5> &tac) const; 1949Â  1952Â  bool setTACOnline(const Array<5> &tac);
```

```
1955Â  bool getTACOnline(Array<5> &tac) const; 1956Â  1959Â  bool setTACDefault(const Array<5> &tac); 1962Â
bool getTACDefault(Array<5> &tac) const; 1963Â  1966Â  bool setThreshold(unsigned th); 1969Â  bool
getThreshold(unsigned &th) const; 1970Â  1973Â  bool setSpecialTransactions(const EMVSpecialTransactions &tx);
1976Â  bool getSpecialTransactions(EMVSpecialTransactions &tx) const; 1977Â  1980Â  bool setLanguage(unsigned
char info); 1983Â  bool getLanguage(unsigned char &info) const; 1984Â  1987Â  bool setTextID(unsigned char id);
1990Â  bool getTextID(unsigned char &id) const; 1991Â  1994Â  bool getBalanceBefore(BCD<6> &amount) const;
1997Â  bool getBalanceBefore(int64_t &amount) const; 1998Â  1999Â  /* deprecated: always returns false */ 2000Â
bool getDefaultAppName(std::string &appname) const; 2001Â  2004Â  bool getApplicationLabel(std::string &appname)
const; 2005Â  2008Â  bool setAmountConfirmation(unsigned char when); 2011Â  bool getAmountConfirmation(unsigned
char &when) const; 2012Â  2015Â  bool setTransactionOptions(const Array<5> &opt); 2018Â  bool
setTransactionOptions(const EMVTransactionOptions &opt) 2019Â  { return setTransactionOptions(opt.array()); }
2022Â  bool getTransactionOptions(Array<5> &opt) const; 2025Â  bool getTransactionOptions(EMVTransactionOptions
&opt) const 2026Â  { return getTransactionOptions(opt.array()); } 2027Â  2030Â  bool setTransactionSteps(const
Array<3> &step); 2033Â  bool setTransactionSteps(const EMVTransactionSteps &step) 2034Â  { return
setTransactionSteps(step.array()); } 2037Â  bool getTransactionSteps(Array<3> &step) const; 2040Â  bool
getTransactionSteps(EMVTransactionSteps &step) const 2041Â  { return getTransactionSteps(step.array()); } 2042Â
2045Â  bool getBalanceAfter(BCD<6> &amount) const; 2048Â  bool getBalanceAfter(int64_t &amount) const; 2049Â
2052Â  bool setDCCInfo(const Array<3> &info); 2055Â  bool getDCCInfo(Array<3> &info) const; 2056Â  2059Â  bool
setFallbackMSROptions(unsigned char opt); 2062Â  bool getFallbackMSROptions(unsigned char &opt) const; 2063Â
2066Â  bool setForceOnline(unsigned char force); 2069Â  bool getForceOnline(unsigned char &force) const; 2070Â
2073Â  bool setPINBypass(bool bypass); 2076Â  bool getPINBypass(bool &bypass) const; 2077Â  2080Â  bool
getEMVStatus(unsigned &status) const; 2081Â  2084Â  bool setForceAccept(unsigned char force); 2087Â  bool
getForceAccept(unsigned char &force) const; 2088Â  2091Â  bool setNoDirectorySelect(unsigned char nodir); 2094Â
bool getNoDirectorySelect(unsigned char &nodir) const; 2095Â  2098Â  bool setPreselectedLanguage(unsigned char
lang); 2101Â  bool getPreselectedLanguage(unsigned char &lang) const; 2102Â  2105Â  bool
setOnlineSwitch(unsigned char online); 2108Â  bool getOnlineSwitch(unsigned char &online) const; 2109Â  2112Â
bool setNoCardholderConfirm(unsigned char conf); 2115Â  bool getNoCardholderConfirm(unsigned char &conf) const;
2116Â  2119Â  bool setEMVPassthrough(unsigned char ep); 2122Â  bool getEMVPassthrough(unsigned char &ep) const;
2123Â  2126Â  bool getPDOLInfo(unsigned char &info) const; 2127Â  2130Â  bool getCardType(unsigned char &type)
const; 2131Â  2134Â  bool setOnlineResult(bool result); 2137Â  bool getOnlineResult(bool &result) const; 2138Â
2141Â  bool setVoiceReferral(bool result); 2144Â  bool getVoiceReferral(bool &result) const; 2145Â  2148Â  bool
setAuthData(const bytevector &data); 2150Â  template<unsigned N> bool setAuthData(const unsigned char (&v)[N])
2151Â  { return setAuthData(bytevector(v,v+N)); } 2154Â  bool getAuthData(bytevector &data) const; 2155Â  2158Â
bool setCriticalScript(const bytevector &script); 2160Â  template<unsigned N> bool setCriticalScript(const
unsigned char (&v)[N]) 2161Â  { return setCriticalScript(bytevector(v,v+N)); } 2164Â  bool
getCriticalScript(bytevector &script) const; 2165Â  2168Â  bool setNonCriticalScript(const bytevector &script);
2170Â  template<unsigned N> bool setNonCriticalScript(const unsigned char (&v)[N]) 2171Â  { return
setNonCriticalScript(bytevector(v,v+N)); } 2174Â  bool getNonCriticalScript(bytevector &script) const; 2175Â
2178Â  bool setAuthResponseReferral(const Array<2> &ac); 2181Â  bool getAuthResponseReferral(Array<2> &ac)
const; 2182Â  2185Â  bool setAuthResponseSwitch(const Array<2> &resp); 2188Â  bool getAuthResponseSwitch(Array<2
> &resp) const; 2189Â  2192Â  bool setAuthResponseWrongPIN(const Array<2> &ac); 2195Â  bool
getAuthResponseWrongPIN(Array<2> &ac) const; 2196Â  2199Â  bool setAuthResponseOnlinePIN(const Array<2> &resp);
2202Â  bool getAuthResponseOnlinePIN(Array<2> &resp) const; 2203Â  2206Â  bool setAdditionalResponseOK(const
Array<2> &ac); 2209Â  bool getAdditionalResponseOK(Array<2> &ac) const; 2210Â  2213Â  bool setWriteDSData(const
bytevector &dsdata); 2216Â  bool getWriteDSData(bytevector &dsdata) const; 2217Â  2220Â  bool
setOfflinePINErrors(unsigned char errors); 2223Â  bool getOfflinePINErrors(unsigned char &errors) const; 2224Â
2227Â  bool setWriteDataStoragedData(const bytevector &data); 2230Â  bool getWriteDataStoragedData(bytevector
&data) const; 2231Â  2234Â  bool setDCCForbidden(unsigned char dcc); 2237Â  bool getDCCForbidden(unsigned char
&dcc) const; 2238Â  2241Â  bool setDomesticInfo(const Array<3> &info); 2244Â  bool getDomesticInfo(Array<3>
&info) const; 2245Â  2248Â  bool setDomesticOptions(const Array<3> &opt); 2251Â  bool setDomesticOptions(const
EMVDomesticOptions &opt) 2252Â  { return setDomesticOptions(opt.array()); } 2255Â  bool getDomesticOptions(Array
<3> &opt) const; 2258Â  bool getDomesticOptions(EMVDomesticOptions &opt) const 2259Â  { return
getDomesticOptions(opt.array()); } 2260Â  2263Â  bool setVelocityOriginalIndex(unsigned char idx); 2266Â  bool
getVelocityOriginalIndex(unsigned char &idx) const; 2267Â  2270Â  bool getErrorData(Array<15> &error) const;
2271Â  2274Â  bool getDisplayTextIndex(unsigned char &id) const; 2275Â  2278Â  bool
setVelocityProcessingResultBitmap(const Array<3> &bm); 2281Â  bool getVelocityProcessingResultBitmap(Array<3>
&bm) const; 2282Â  2285Â  bool getKernelDebug(bytevector &debug) const; 2286Â  2289Â  bool setExcludeAID(const
std::vector<bytevector> &aid); 2292Â  bool getExcludeAID(std::vector<bytevector> &aid) const; 2293Â  2301Â  bool
setExcludeCombos(const std::vector<std::pair<bytevector, Array<3> > > &combos); 2302Â  2305Â  bool
setCardholderInfo(unsigned char info); 2308Â  bool getCardholderInfo(unsigned char &info) const; 2309Â  2312Â
bool setMerchantInfo(unsigned char info); 2315Â  bool getMerchantInfo(unsigned char &info) const; 2316Â  2319Â
bool setConfirmAmount(bool confirm); 2322Â  bool getConfirmAmount(bool &confirm) const; 2323Â  2326Â  bool
setPANInBlacklist(unsigned char contained); 2329Â  bool getPANInBlacklist(unsigned char &contained) const;
2330Â  2333Â  bool setStoredAmount(const BCD<6> &amount); 2336Â  bool setStoredAmount(int64_t amount); 2339Â
bool getStoredAmount(BCD<6> &amount) const; 2342Â  bool getStoredAmount(int64_t &amount) const; 2343Â  2346Â
bool setSelector(unsigned char sel); 2349Â  bool getSelector(unsigned char &sel) const; 2350Â  2353Â  bool
setReducedCanditateList(const bytevector &cl); 2355Â  template<unsigned N> bool setReducedCanditateList(const
unsigned char (&v)[N]) 2356Â  { return setReducedCanditateList(bytevector(v,v+N)); } 2359Â  bool
```

```
getReducedCanditateList(bytevector &cl) const; 2360Â  2363Â  bool setModifyTransaction(unsigned char m); 2366Â
bool getModifyTransaction(unsigned char &m) const; 2367Â  2370Â  bool setSkipPostprocessing(unsigned char skip);
2373Â  bool getSkipPostprocessing(unsigned char &skip) const; 2374Â  2377Â  bool setPINInfo(unsigned char info);
2380Â  bool getPINInfo(unsigned char &info) const; 2381Â  2384Â  bool setPINPublicModulus(const bytevector
&modulus); 2386Â  template<unsigned N> bool setPINPublicModulus(const unsigned char (&v)[N]) 2387Â  { return
setPINPublicModulus(bytevector(v,v+N)); } 2390Â  bool getPINPublicModulus(bytevector &modulus) const; 2391Â
2394Â  bool setPINPublicExponent(unsigned modulus); 2397Â  bool getPINPublicExponent(unsigned &modulus) const;
2398Â  2401Â  bool setDCCMode(unsigned char mode); 2404Â  bool getDCCMode(unsigned char &mode) const; 2405Â
2408Â  bool setDomesticAppResult(unsigned char res); 2411Â  bool getDomesticAppResult(unsigned char &res) const;
2412Â  2415Â  bool setKernelToUse(const Array<3> &kernel); 2418Â  bool getKernelToUse(Array<3> &kernel) const;
2419Â  2422Â  bool setCommunicationError(unsigned char err); 2425Â  bool getCommunicationError(unsigned char
&err) const; 2426Â  2429Â  bool getDestination(int &dest) const; 2430Â  2433Â  bool
setDataExchangeRequest(unsigned char req); 2436Â  bool getDataExchangeRequest(unsigned char &req) const; 2437Â
2440Â  bool setDataExchangeState(unsigned char state); 2443Â  bool getDataExchangeState(unsigned char &state)
const; 2444Â  2447Â  bool setDataExchangeMode(unsigned char state); 2450Â  bool getDataExchangeMode(unsigned
char &state) const; 2451Â  2454Â  bool setDataExchangeDataNeeded(const bytevector &dn); 2456Â  template<unsigned
N> bool setDataExchangeDataNeeded(const unsigned char (&v)[N]) 2457Â  { return
setDataExchangeDataNeeded(bytevector(v,v+N)); } 2460Â  bool getDataExchangeDataNeeded(bytevector &dn) const;
2461Â  2464Â  bool setModifiedCandidates(const bytevector &cand); 2466Â  template<unsigned N> bool
setModifiedCandidates(const unsigned char (&v)[N]) 2467Â  { return setModifiedCandidates(bytevector(v,v+N)); }
2470Â  bool getModifiedCandidates(bytevector &cand) const; 2471Â  2474Â  bool getSetupResult(Array<5> &result)
const; 2475Â  2478Â  bool setCBDisplayTextIndex(unsigned char idx); 2481Â  bool getCBDisplayTextIndex(unsigned
char &idx) const; 2482Â  2485Â  bool getCardID(Array<20> &id) const; 2486Â  2489Â  bool getTrackStatus(Array<3>
&track_status) const; 2490Â  2493Â  bool getObfuscatedTrack1String(std::string &track) const; 2494Â  2497Â  bool
getTecselData(bytevector &data) const; 2498Â  2501Â  bool setKernelID(const Array<3> &id); 2504Â  bool
setKernelID(unsigned id); 2507Â  bool getKernelID(Array<3> &id) const; 2510Â  bool getKernelID(unsigned &id)
const; 2511Â  2514Â  bool setAppNameChosen(const std::string &name); 2517Â  bool getAppNameChosen(std::string
&name) const; 2518Â  2521Â  bool setApplicationKernelId(const Array<3> &id); 2524Â  bool
setApplicationKernelId(unsigned id); 2527Â  bool getApplicationKernelId(Array<3> &id) const; 2530Â  bool
getApplicationKernelId(unsigned &id) const; 2531Â  2534Â  bool setAppNameUtf8(const std::string &name); 2537Â
bool getAppNameUtf8(std::string &name) const; 2538Â  2541Â  bool getKernelPrivateNonBER(bytevector &nber) const;
2542Â  2544Â  // constructed data elements // 2546Â  2549Â  bool setFCIIssuerDiscrData(const TLV &data); 2552Â
bool getFCIIssuerDiscrData(TLV &data) const; 2553Â  2556Â  bool setMemorySlotUpdateTemplate(const TLV &t);
2559Â  bool getMemorySlotUpdateTemplate(TLV &t) const; 2560Â  2563Â  bool setMemorySlotReadTemplate(const TLV
&t); 2566Â  bool getMemorySlotReadTemplate(TLV &t) const; 2567Â  2569Â  struct DomesticApp { 2570Â  bytevector
aid; 2571Â  std::string appLabel; 2572Â  unsigned char asi; 2573Â  unsigned char specialTX[8]; 2574Â  unsigned
char noEMVConformSelect; 2575Â #if __cplusplus >= 201103 2576Â  DomesticApp(const bytevector &aid_, 2577Â  const
std::string &appLabel_, 2578Â  unsigned char asi_, 2579Â  const unsigned char specialTx_[8], 2580Â  unsigned
char noEMVConformSelect_) 2581Â  : aid(aid_), 2582Â  appLabel(appLabel_), 2583Â  asi(asi_), 2584Â
noEMVConformSelect(noEMVConformSelect_) { 2585Â  memcpy(specialTX, specialTx_, 8); 2586Â  } 2587Â  DomesticApp()
= default; 2588Â #endif 2589Â  }; 2590Â  2593Â  bool setDomesticApps(const std::vector<DomesticApp> &app);
2596Â  bool getDomesticApps(std::vector<DomesticApp> &app) const; 2597Â  2599Â  struct FallbackMSR { 2600Â
unsigned char mid; 2601Â  unsigned char specialTX[8]; 2602Â  unsigned char fallback; 2603Â  unsigned char
options; 2604Â #if __cplusplus >= 201103 2605Â  FallbackMSR(unsigned char mid_, 2606Â  const unsigned char
*specialTx_, 2607Â  unsigned char fallback_, 2608Â  unsigned char options_) 2609Â  : mid(mid_),
fallback(fallback_), options(options_) { 2610Â  memcpy(specialTX, specialTx_, 8); 2611Â  } 2612Â  FallbackMSR()
= default; 2613Â #endif 2614Â  }; 2615Â  2618Â  bool setFallbackMSR(const std::vector<FallbackMSR> &mid); 2621Â
bool getFallbackMSR(std::vector<FallbackMSR> &mid) const; 2622Â  2624Â  struct Candidate { 2625Â  bytevector
aid; 2626Â  std::string appName; 2627Â #if __cplusplus >= 201103 2628Â  Candidate(const bytevector &aid_, const
std::string &appName_) 2629Â  : aid(aid_), appName(appName_) {} 2630Â  Candidate() = default; 2631Â #endif
2632Â  }; 2633Â  2636Â  bool setCandidateList(const std::vector<Candidate> &cand); 2639Â  bool
getCandidateList(std::vector<Candidate> &cand) const; 2640Â  2642Â  struct CandidateExt { 2643Â  bytevector aid;
2644Â  std::string appName; 2645Â  std::string language; 2646Â  unsigned kernelID; 2647Â  unsigned char
velocityOrigIdx; 2648Â  unsigned char velocityProcResult; 2649Â #if __cplusplus >= 201103 2650Â
CandidateExt(const bytevector &aid_, 2651Â  const std::string &appName_, 2652Â  const std::string &language_,
2653Â  unsigned kernelID_, 2654Â  unsigned char velocityOrigIdx_, 2655Â  unsigned char velocityProcResult_)
2656Â  : aid(aid_), 2657Â  appName(appName_), 2658Â  language(language_), 2659Â  kernelID(kernelID_), 2660Â
velocityOrigIdx(velocityOrigIdx_), 2661Â  velocityProcResult(velocityProcResult_) {} 2662Â  CandidateExt() =
default; 2663Â #endif 2664Â  }; 2665Â  2668Â  bool setCBCandidateList(const std::vector<CandidateExt> &cand);
2671Â  bool getCBCandidateList(std::vector<CandidateExt> &cand) const; 2672Â  2675Â  bool
setDataExchangeReceivedData(const TLV &data); 2678Â  bool getDataExchangeReceivedData(TLV &data) const; 2679Â
2682Â  bool setDataExchangeSendData(const TLV &data); 2685Â  bool getDataExchangeSendData(TLV &data) const;
2686Â  2689Â  bool getNFCVASData(TLV &data) const; 2690Â }; 2691Â  2692Â } // namespace vfisdi 2693Â
2694Â #undef DllSpec 2695Â  2696Â #endif
```