

Secure Electronic Transaction (SET) Supplement for the
v1.0 Internet Open Trading Protocol (IOTP)

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document describes detailed Input/Output parameters for the Internet Open Trading Protocol (IOTP) Payment Application Programming Interface (API). It also describes procedures in the Payment Bridge for the use of SET (SET Secure Electronic Transaction) as the payment protocol within Version 1.0 of the IOTP.

Table of Contents

1. Introduction.....	3
1.1 Objectives of this Document.....	3
1.2 Scope of this specification.....	3
1.2.1 The version of IOTP specification.....	3
1.2.2 The version of SET specification.....	4
1.2.3 The version of IOTP Architecture document.....	4
1.3 Audience.....	4
1.4 Notation.....	4
1.5 Terminology.....	4
2. Requirements & Development Policy.....	4
3. Business Models.....	5
3.1 Entity models between SET and IOTP.....	5
3.2 Role of Participants.....	5
3.3 Scope of Transaction Types.....	6
3.4 Types of transaction not in scope.....	6
4. Architecture of SET/IOTP.....	7
5. Trading Types of SET/IOTP.....	7
5.1 Baseline Purchase.....	7
5.2 Cash Advances.....	8
5.3 Status Inquiry	8

6. General Flow of SET/IOTP.....	8
6.1 Baseline Purchase.....	9
6.1.1 Brand Independent Baseline Purchase.....	9
6.1.2 Brand Dependent Baseline Purchase.....	13
6.2 Cash Advances.....	14
6.3 Status Inquiry.....	15
7. IOTP Payment APIs.....	16
7.1 Brand Compilation Related API Calls.....	16
7.1.1 Find Accepted Payment Brand.....	16
7.1.2 Find Accepted Payment Protocol.....	17
7.1.3 Get Payment Initialization Data.....	18
7.1.4 Inquire Authentication Challenge.....	19
7.1.5 Authenticate.....	19
7.1.6 Check Authentication Response.....	19
7.2 Brand Selection Related API Calls.....	20
7.2.1 Find Payment Instrument.....	20
7.2.2 Check Payment Possibility.....	21
7.3 Payment Transaction Related API Calls.....	22
7.3.1 Start Payment Consumer.....	22
7.3.2 Start Payment Payment Handler.....	23
7.3.3 Resume Payment Consumer.....	24
7.3.4 Continue Process.....	25
7.3.5. Change Process State.....	26
7.4 General Inquiry API Calls.....	26
7.4.1 Payment Instrument Inquiry.....	26
7.4.2 Inquire Pending Payment.....	26
7.4.3 Remove Payment Log.....	27
7.5 Payment Related Inquiry API Calls.....	27
7.5.1 Check Payment Receipt.....	27
7.5.2 Expand Payment Receipt.....	27
7.5.3 Inquire Process State.....	28
7.5.4 Start Payment Inquiry.....	29
7.5.5 Inquire Payment Status.....	30
8. SET dependent Process.....	30
8.1 Relationships between them for IOTP Purchase/Cash Advances.....	30
8.2 Definition of Identifiers.....	31
8.2.1 Definition of BrandId.....	31
8.2.2 Definition of ProtocolBrandId.....	31
8.2.3 Definition of ProtocolId.....	33
8.2.4 Relationship between Ids.....	33
8.3 Process prior to Payment.....	34
8.3.1 FindAcceptedPaymentProtocol Function.....	34
8.3.2 FindPaymentInstrument Function.....	35
8.3.3 GetPaymentInitializationData Function.....	36
8.4 Process of Payment.....	37
8.4.1 StartPaymentConsumer Function.....	37
8.4.2 StartPaymentPaymentHandler Function.....	41
8.4.3 ContinueProcess Function (Consumer Side).....	42

8.4.4 ContinueProcess Function (Payment Handler Side).....	43
8.4.5 InquireProcessState Function.....	45
8.5 Payment Receipt.....	45
8.5.1 CheckPayReceipt Function.....	45
8.5.2 ExpandPayReceipt Function.....	45
8.6 Status Inquiry.....	46
8.7 Resume Process.....	47
8.8 SET Scheme Specific Authentication on IOTP.....	47
8.9 SET Bridge ProcessState.....	48
8.9.1 SET Bridge ProcessState of Consumer.....	48
8.9.2 SET Bridge ProcessState of Payment Handler.....	49
8.10 Relationship between Pay Step and Deliv Step on SET/IOTP..	49
8.11 Completion Code.....	50
8.12 PercentComplete.....	50
8.13 Severity.....	51
9. Error Handling.....	51
9.1 Types of Errors.....	51
9.2 IOTP Level Error (OAC Error).....	52
9.3 IOTP Level Error (SET Bridge Error).....	52
9.4 SET Level Error (SET Technical Error).....	52
9.4.1 SET Initiation Error.....	52
9.4.2 SET Transaction Error.....	53
9.5 SET Level Error (SET Business Error).....	53
10. Security Considerations.....	54
11. References.....	54
12. IANA Considerations.....	55
13. Acknowledgement.....	55
14. Author's Address.....	55
15. Full Copyright Statement.....	56

1. Introduction

This chapter describes the outline of this document.

1.1 Objectives of this Document

This document describes how SET (SET Secure Electronic Transaction) works within the IOTP (Internet Open Trading Protocol).

1.2 Scope of this specification

1.2.1 The version of IOTP specification

This document is written based on IOTP Version 1.0 [RFC 2801].

1.2.2 The version of SET specification

This document is written based on SET Version 1.0 [SET].

1.2.3 The version of IOTP Architecture document

This document is written based on IOTP Payment API document Version 1.0 [IOTP Payment API].

1.3 Audience

This document is indented for readers who are familiar with the following documents:

- 1) IOTP Specification Version 1.0 [RFC 2801]
- 2) SET Specification, in particular Book 2:Programmer's Guide and Book3:Formal Protocol Definition,
- 3) External Interface Guide to SET Secure Electronic Transaction
- 4) Internet Open Trading Supplement: Architecture and Payment API [IOTP API]

1.4 Notation

SET Messages and Elements are described with the prefix "SET".

Examples:

SET PRes

SET OD

SET SaleDetail

1.5 Terminology

This document uses the following terms:

SET/IOTP	The specification described in this document.
SET related message	Both SET Messages and SET Initiation Messages

2. Requirements & Development Policy

This chapter describes the requirements and development policies of SET/IOTP.

The requirements of SET/IOTP are as follows:

- o To be based on SET specifications. Interoperability at the payment level must be maintained.

- o To not enforce modifications which are specific to SET/IOTP. General features of IOTP should not be tampered with to cater to a particular payment method.
- o To keep integrity between IOTP and SET. Inconstancy must not be raised between IOTP and SET elements when they have the same meaning.

The development policy of SET/IOTP is as follows:

- o To minimize the number of message round trips
- o To minimize the length of messages

3. Business Models

This chapter describes the difference in entity models between SET and IOTP, the definitions of Trading Roles in SET/IOTP, and the scope of SET/IOTP.

3.1 Entity models between SET and IOTP

The following table describes how SET and IOTP entities correspond to each other.

IOTP Entity		SET Entity

Consumer	<--->	Card Holder
Merchant	<--->	Merchant (Initiation)
Payment Handler	<--->	Merchant (Payment)
Delivery Handler	<--->	None
None	<--->	Acquirer

Figure 1 Entity Models between SET and IOTP

3.2 Role of Participants

The following table describes the trading roles in SET/IOTP.

Trading Roles	Role

Consumer	An Individual who purchases goods and/or services, and pays for the value received by choosing a SET Transaction. This individual corresponds with the CardHolder in SET.

Merchant	An organization that provides goods and/or services for purchase, accepts payment methods, delivers invoices and triggers payment processes.
Payment Handler	An organization that processes negotiations on payments including SET payment transactions.
Delivery Handler	An Organization that ships digital or physical goods to the Consumer.
Customer Care Provider	The same as in [RFC 2801].
Merchant Care Provider	The same as in [RFC 2801].

3.3 Scope of Transaction Types

The types of IOTP transactions that are supported in this document are as follows:

- o Brand Independent Baseline Purchase when SET is used for payment
- o Brand Dependent Baseline Purchase when SET is used for payment
- o Cash Advances (Brand Independent and Brand Dependent case)
- o Status Inquiry on SET payments

3.4 Types of transaction not in scope

The types of transactions that are NOT covered in this document are as follows:

- o Credit Reversal Process
- o Customer Care Service with Consumer Related SET Certificate Registration
- o Customer Care Service with Consumer Related SET Certificate Registration Inquiry

4. Architecture of SET/IOTP

SET/IOTP Architecture is as follows:

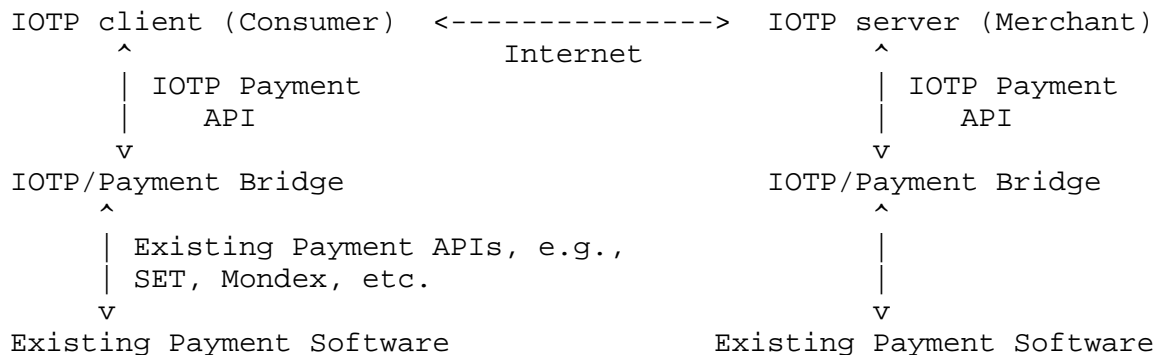


Figure 2 SET/IOTP Architecture

IOTP Application Core (OAC): Software that processes IOTP messages.
 IOTP Payment Bridge (OPB): Interface between OAC and Existing Payment Software. SET Bridge is also an interface between OAC and SET Core.

Existing Payment Software (EPS): Existing Software that processes Payments. The SET Core is software that supports mechanisms in SET specification from Book1 to Book3. EPS does NOT necessarily have to implement the SET Initiation Processor, which is specified in SET FIG. SET Related Module Both SET related OPB and EPS.

5. Trading Types of SET/IOTP

This chapter describes the outline of SET/IOTP trading types.

5.1 Baseline Purchase

Three steps will take place in a Baseline Purchase in the following order:

(1) Offer Step

Consumer selects goods/services over the Internet, for instance on the web, and then chooses the payment method (SET is selected), the SET brand, the payment currency, and then confirms the invoice.

There are two Offer Process types, Brand Independent and Brand Dependent.

(1-a) Brand Independent Purchase

In a Brand Independent Purchase, the Merchant sends the TPO Block and Offer Response Block simultaneously after the consumer's purchase decision. The Brand Independent Purchase has the merit of eliminating one round of messages compared with the Brand Dependent Purchase because the contents of the Offer Response Block (for example, the description on the invoice) do not change based on the selected brand.

(1-b) Brand Dependent Purchase

Brand Dependent Purchase is used when the contents of the Offer Response Block are dependent on the selected Payment Brand. With this method, the currency selection and discounts based on payment method can be implemented.

(2) Payment Step

The Consumer confirms the order and then pays for the order with a SET Transaction. The SET Transaction messages will be encapsulated in IOTP Messages.

(3) Delivery Step

After completing the Payment, the Consumer receives the goods/services via either on-line or physical delivery.

5.2 Cash Advances

Cash Advances can be made via a Value Exchange Transaction in IOTP. A first Payment by SET and a second Payment by some other payment mechanism is supported in Baseline IOTP. The Cash Advance has two types - Brand Independent and Brand Dependent Cases.

5.3 Status Inquiry

A Consumer can send a SET Payment Inquiry in IOTP. The SET Message is encapsulated in an IOTP Message.

6. General Flow of SET/IOTP

This chapter illustrates the general SET/IOTP message flows.

6.1 Baseline Purchase

Baseline purchases consist of two types, Brand Independent Purchase and Brand Dependent Purchase. Each type is illustrated in the charts below.

6.1.1 Brand Independent Baseline Purchase

The general flow of a Brand Independent Purchase is as follows:

(1) Consumer Side (Before PayRequest Message)

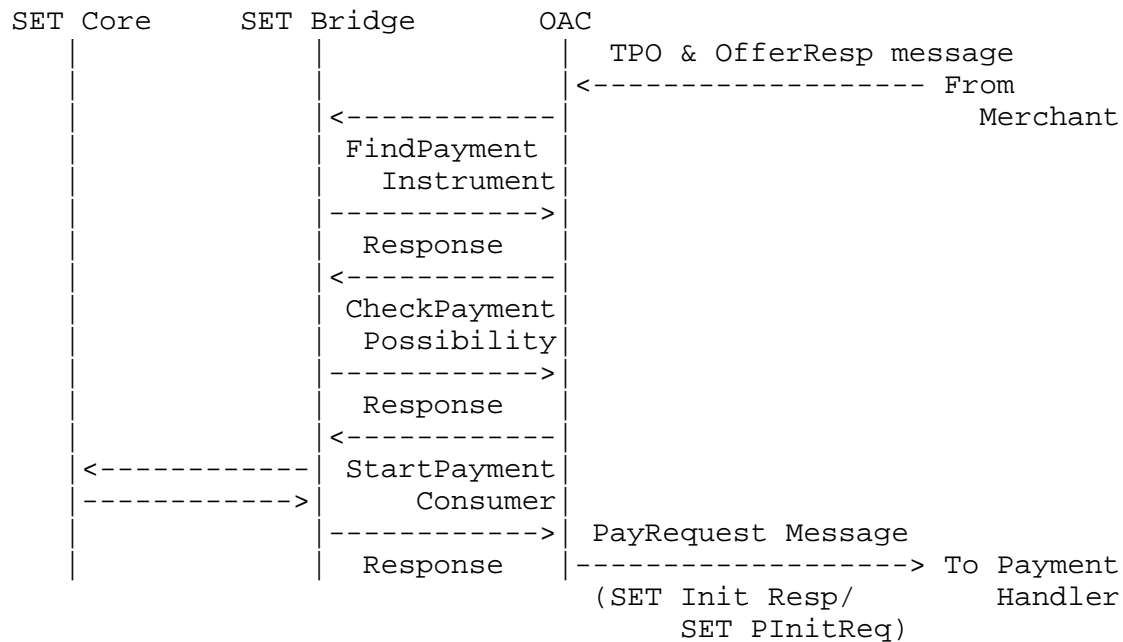


Figure 3 Consumer Side for Brand Independent (1)

(2) Consumer Side (After PayRequest Message)

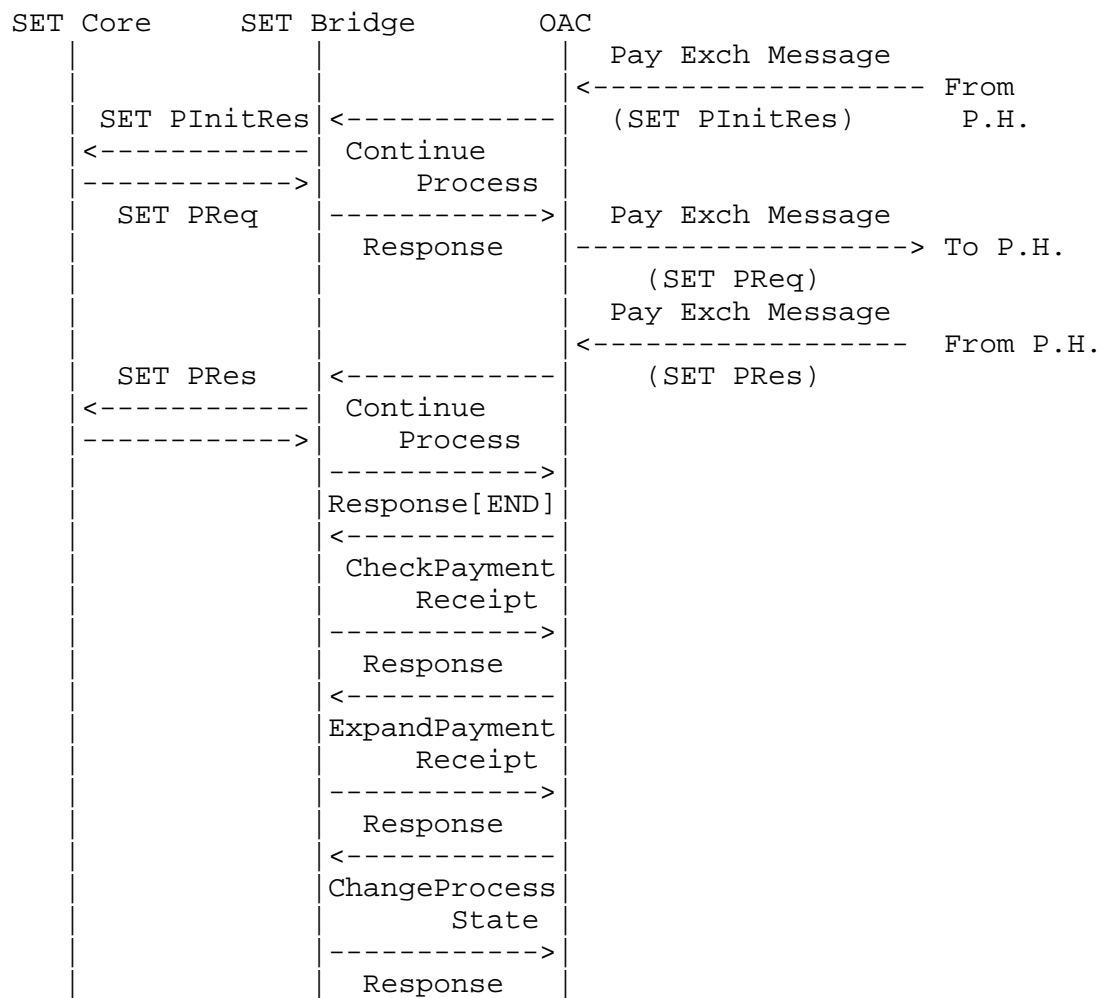


Figure 4 Consumer Side flow for Brand Independent (2)

(3) Merchant Side

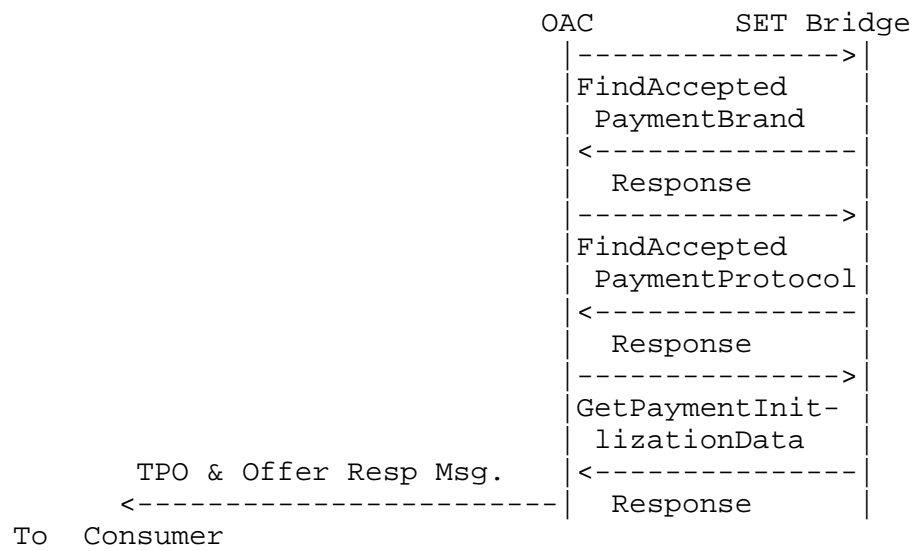


Figure 5 Merchant Side flow for Brand Independent

(4) Payment Handler Side.

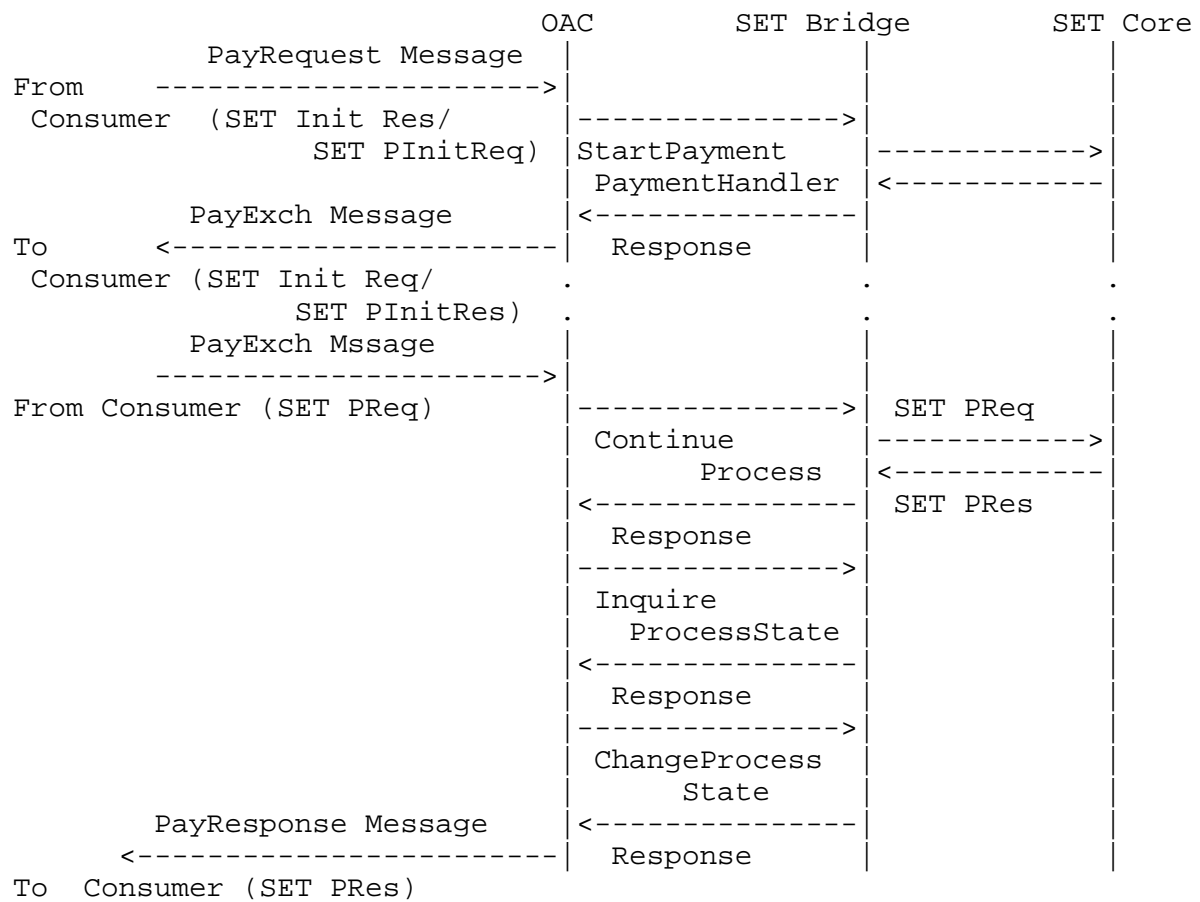


Figure 6 Payment Handler side flow for Brand Independent

6.1.2 Brand Dependent Baseline Purchase

The general flow of a Brand Dependent Purchase is as follows:

(1) Consumer Side (Before PayRequest Message)

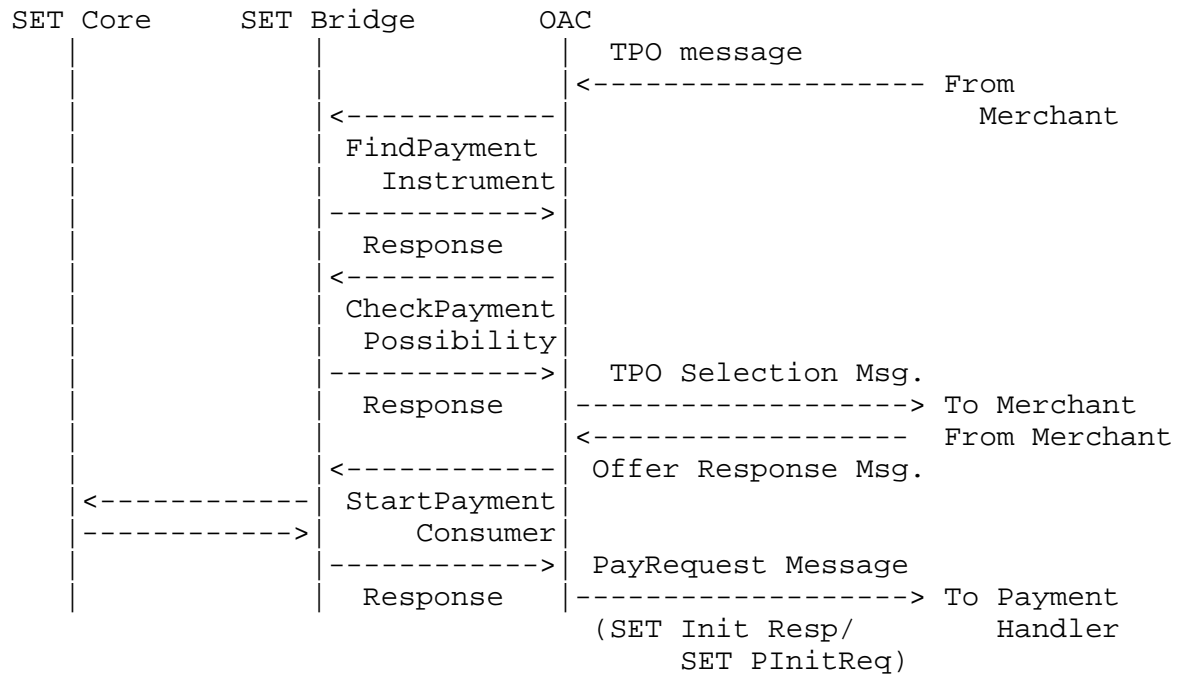


Figure 7 Consumer Side flow for Brand Dependent (1)

(2) Consumer Side (After PayRequest Message)

This flow is the same as Brand Independent.

(3) Merchant Side

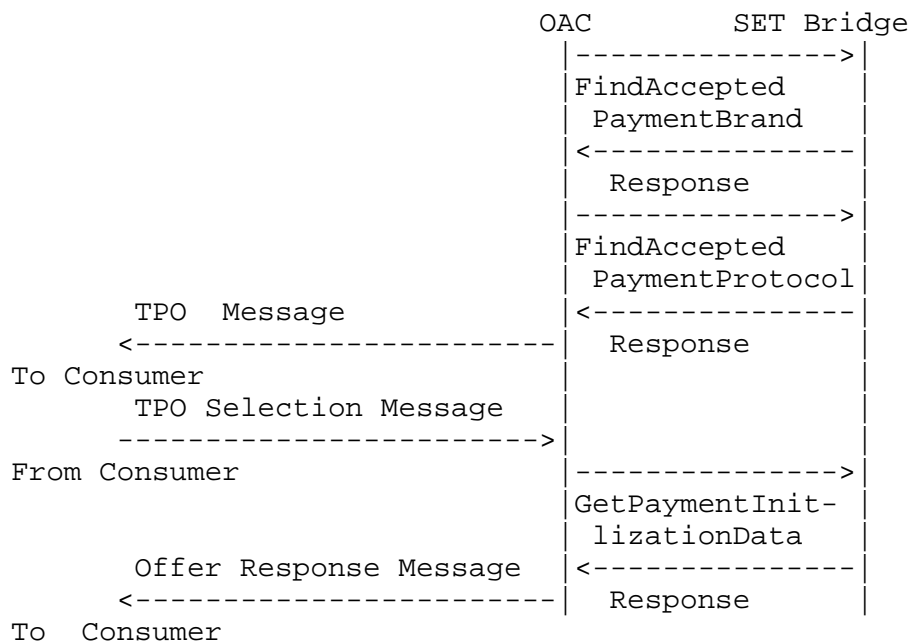


Figure 8 Merchant Side flow for Brand Dependent (1)

(4) Payment Handler Side

This flow is the same as Brand Independent.

6.2 Cash Advances

IOTP Cash Advances processes can be made with a credit card using an IOTP Value Exchange Transaction. In Cash Advances a first Payment by a SET Transaction, and a second Payment by some other payment mechanism, is supported in Baseline IOTP. The general flow is omitted.

6.3 Status Inquiry

The general flow of a Status Inquiry is as follows:

(1) Consumer Side

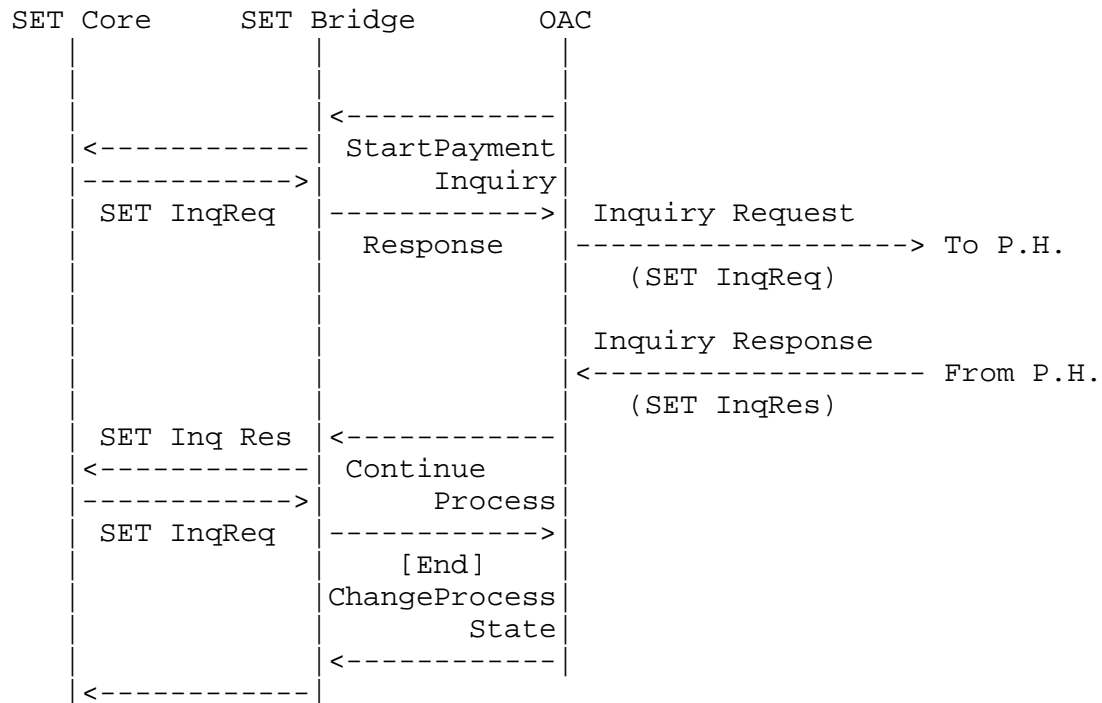


Figure 9 Consumer Side flow for Status Inquiry

(2) Payment Handler Side

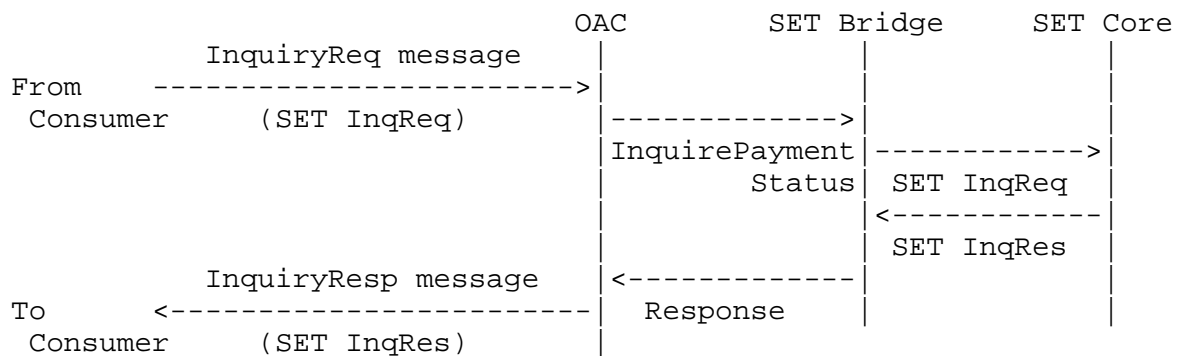


Figure 10 Payment Handler Side flow for Status Inquiry

7. IOTP Payment APIs

This section provides a summary of SET/IOTP interactions with API calls as in [IOTP Payment API].

The description of parameters hereafter are written as follows:

Parameter name : Mandatory (M) or Optional (O) : Description

For more details on the IOTP Payment APIs, see [IOTP Payment API].
"-" in the Description is the same as description in the [IOTP Payment API].

Notice: Status is the status of SET/IOTP. Though some Fields are specified "#IMPLIED" in [IOTP Payment API], if the fields must be used in SET/IOTP, this document specifies the status as Mandatory, (M).

7.1 Brand Compilation Related API Calls

7.1.1 Find Accepted Payment Brand

Receive the payment scheme specific packaged data to generate Brand Component. In this version of SET/IOTP, This API must be called before Find Accepted Payment Protocol function.

Input Parameters

PayDirection	: M	: This must be set "Debit".
CurrCodeType	: M	: This should be set "ISO4217-A".
CurrCode	: M	: -
Amount	: M	: -
MerchantPayId	: M	: -
MerchantOrgId	: M	: -
WalletId	: O	: -
MerchantData	: O	: The details are not specified in this document.

Output Parameters

BrandItem	: M	: See NOTE below.
-----------	-----	-------------------

NOTE: Parameters of BrandItem

```

-----
BrandId          : M : This is defined in the section 8.2.1.
xml:lang         : M : -
BrandName        : M : Brand Name, such as "MasterCard".
BrandLogoNetLocn : M : -
BrandNarrative   : O : This is not specified in this document.
BrandPackaged    : O : This is not used in the SET/IOTP.
Content

```

7.1.2 Find Accepted Payment Protocol

Receive the payment scheme specific packaged data to generate the PayProtocol Component.

Input Parameters

```

-----
BrandId          : M : This is defined in the section 8.2.1.
PayDirection     : M : This must be set "Debit".
CurrCodeType     : M : This should be set "ISO4217-A".
CurrCode         : M : -
Amount           : M : -
MerhcantPayId    : M : -
MercahntOrgId    : M : -
WalletId         : O : -
BrandPackaged    : O : This is not used in the SET/IOTP.
Content
MerchantData     : O : This is not specified in the SET/IOTP.

```

Output Parameters

```

-----
ProtocolItem     : M : See NOTE below.
BrandItem        : M : -

```

NOTE Parameters of ProtocolItem

```

-----
ProtocolId       : M : This is set "SETv1.0".
ProtocolBrandId  : M : This is set the Payment Protocol Specific
                        ID corresponding to the BrandId as Input
                        Parameter and ProtocolId as the
                        Output Parameter. For the detail,
                        see 8.2.2.
xml:lang         : M : -
ProtocolName     : M : This is not specified in this document
                        but must be included the protocol name
                        and its version at least.

```

PayReqNetLocn : O : The Net Location indicating where a
unsecured Payment Request Message should
be sent if this protocol choice is used.

SecPayReqNetLocn : O : The Net Location indicating where
a secured Payment Request Message
should be sent if this protocol choice
is used.

ProtocolAmount : O : This is not used in the SET/IOTP.
PackagedContent

PayProtocol : M : The XML Packaged Data, which includes
PackagedContent the information for the 1st SET
Initiation Process. See for the details
to section 8.3.1.

Brand : M : In this document, BrandId, which is the
same as Input Parameter, must be set ONLY.
See NOTE below.

CurrencyAmount : M : See NOTE below.

ProtocolBrand : M : Multiple Components are not arrowed in
the current version of SET/IOTP.

Note Parameters of CurrencyAmount

CurrCodeType : M : This should be set "ISO4217-A".

CurrCode : M : -

Amount : M : -

Note Parameters of Brand

BrandId : M : -

7.1.3 Get Payment Initialization Data

This API is used to get the packaged content in Payment Component.

Input Parameters

BrandId : M : See the details of section 8.2.1.

MerchantPayId : M : -

PayDirection : M : This is set "Debit".

CurrCodeType : M : This is set "ISO5217-A".

CurrCode : M : -

Amount : M : -

OkFrom : M : -

OkTo : M : -

ReceiverOrgId : M : Organization ID which is used to get
TradingRolePackagedContents, which
depend on the organizations for each.

MerchantOrgId : M : -

ProtocolId : M : This field must be set "SETv1.0".
 WalletId : O : -
 PassPhrase : O : -
 ProtocolBrand : M : -
 BrandPackaged : O : This is not used in the current version
 Content of SET/IOTP.
 ProtocolAmount : O : This is not used in the current version
 PackagedContent of SET/IOTP.
 PayProtocolPackaged: M : This field is copied from the
 Content PayProtocol Component.
 OrderPackaged : M : Packaged Data regarding the Order data,
 Content which the Merchant's OAC sets.
 BrandSelBrandInfo : O : This is not used in the current
 PackagedContent version of SET/IOTP.
 BrandSelProtocol : O : This is not used in the
 AmountInfoPackaged current version of SET/IOTP.
 Content
 BrandSelCurrency : O : This is not used in the
 AmountInfo current version of SET/IOTP.
 PackagedContent

Output Parameters

OkFrom : M : -
 OkTo : M : -
 OrderPackaged : M : Changed OrderPackagedContent if
 Content it rewrites the order information.
 Otherwise, passed the same input
 data to OAC.
 TradingRole : O : The receiver depended
 PackagedContent TradingRolePackagedContent. The Name
 Attribute of the packaged contents
 must include "Payment:" as the prefix,
 for example "Payment:SET-OD". Multiple
 TradingRoleData may be returned.

7.1.4 Inquire Authentication Challenge

This is not used in the current version of SET/IOTP.

7.1.5 Authenticate

This is not used in the current version of SET/IOTP.

7.1.6 Check Authentication Response

This is not used in the current version of SET/IOTP.

7.2 Brand Selection Related API Calls

7.2.1 Find Payment Instrument

This API is used to get the Payment Instruments that can be accepted by the Payment Handler on behalf of the Merchant.

Input Parameters

BrandId	: M	: See the details of section 8.2.2.
ProtocolId	: M	: This must be set "SETv1.0".
PayDirection	: M	: This must be set "Debit".
CurrCodeType	: M	: This should be set "ISO5217-A".
CurrCode	: M	: -
Amount	: M	: -
ConsumerPayId	: M	: -
WalletId	: O	: -
ProtocolBrand	: M	: -
BrandPackaged Content	: O	: This is not used in the current version of SET/IOTP.
ProtocolAmount PackagedContent	: O	: This is not used in the current version of SET/IOTP.
PayProtocolPackaged: Content	: M	: See details for section 8.3.1.

Output Parameters

PayInstrument	: M	: Multiple PayInstrument Ids may be returned. See NOTE below.
---------------	-----	--

NOTE Parameters of PayInstrument

Id	: M	: This must be unique each SET Certificates which the Consumer can use.
xml:lang	: M	: -
PayInstName	: M	: -

7.2.2 Check Payment Possibility

If the SET Bridge receives this API Message, the SET Bridge returns three packaged content fields.

Input Parameters

```

-----
BrandId          : M : This is set the consumer selected
                  :   : BrandId.
PaymentInstrumentId: M : This is set the consumer selected
                  :   : PaymentInstrumentID.
PayDirection     : M : This is set "Debit".
CurrCodeType     : M : This is set "ISO4217-A".
CurrCode         : M : -
Amount          : M : -
ProtocolId       : M : This must be set "SETv1.0".
WalletId        : O : -
Passphrase       : O : -
ConsumerPayId    : M : -
ProtocolBrand    : M : This is set the consumer selected
                  :   : ProtocolBrand Component.
BrandPackagedContent : O : This is not used in the current
                  :   : version of SET/IOTP.
ProtocolAmountPackagedContent : O : This is not used in the current
                  :   : version of SET/IOTP.
PayProtocol      : M : This field is copied from the PayProtocol
PackagedContent  :   : Component
  
```

Output Parameter

```

-----
BrandSelBrandInfo : O : This is not used in the current
PackagedContent   :   : version of SET/IOTP.
BrandSelProtocol  : O : This is not used in the
AmountInfoPackagedContent :   : current version of SET/IOTP.

BrandSelCurrency   : O : This is not used in the
AmountInfoPackagedContent :   : current version of SET/IOTP.
  
```

7.3 Payment Transaction Related API Calls

7.3.1 Start Payment Consumer

In SET/IOTP, this API is used for the Consumer's SET Bridge to process the 1st SET Initiation and any subsequent SET messages.

Input Parameters

```

-----
BrandId          : M : ID for the consumer selected
                  :   : Brand. See the details of
                  :   : section 8.2.1.
PaymentInstrumentId: M : ID for the consumer selected
                  :   : Instrument.
CurrCodeType     : M : The consumer selected CurrCodeType.
CurrCode         : M : The consumer selected CurrCode.
Amount          : M : The consumer selected Amount.
PayDirection     : M : Indicates the payment direction
                  :   : from the Consumer's prospective.
ProtocolId       : M : The consumer selected ProtocolId.
OkFrom           : M : -
OkTo             : M : -
ConsumerPayId    : M : -
WalletID         : O : -
Passphrase       : O : -
CallBackFunction : O : This is not used in the SET/IOTP.
CallBackLanguage : O : This is not used in the SET/IOTP.
List
ProtocolBrand    : M : ID for the consumer selected
                  :   : Protocol dependent Brand information.
BrandPackaged    : O : This is not used in the current
Content          :   : version of SET/IOTP.
ProtocolAmount   : O : This is not used in the current
PackagedContent  :   : version of SET/IOTP.
PayProtocolPackaged : M : See section 8.2.2.
Content

```

Output Parameters

```

-----
ContStatus       : M : "Continue" must be set if there is
                  :   : in no problem
PaySchemePackaged : M : See section 6.5.1.
Content

```

7.3.2 Start Payment Payment Handler

This API is used to initiate a payment on the Payment Handler's side. The SET Related Module does a payment initialization. The SET Related Module processes SET Message received and returns the appropriate SET Message (e.g., 2nd SET Initiation or SET PinitRes message).

Input Parameters

```

-----
BrandId          : M : ID for the consumer selected Brand.
                  :   : See the details of section 8.2.1.
ConsumerPayId    : O : ID for the consumer generated payment
                  :   : transaction.
CurrCodeType     : M : The consumer selected CurrCodeType.
                  :   : This should be set "ISO4217-A".
CurrCode         : M : The consumer selected CurrCode.
Amount          : M : The consumer selected Amount.
PayDirection     : M : This is set "Debit".
ProtocolId       : M : The consumer selected ProtocolId.
                  :   : This must be set "SETv1.0".
OkFrom:         : M : -
OkTo            : M : -
PaymentHandlerPayId: M : -
MerchantOrgId    : M : -
WalletID        : O : -
Passphrase      : O : -
CallbackFunction : O : This is not used in the SET/IOTP.
CallbackLanguage : O : This is not used in the SET/IOTP.
List
BrandPackaged    : O : This is not used in the current
Content          :   : version of SET/IOTP.
ProtocolAmountP  : O : This is not used in the current
PackagedContent  :   : version of SET/IOTP.
PayProtocolPackaged: M : -
Content
ProtocolBrand    : M : Information for the consumer selected
                  :   : Protocol dependent Brand.
BrandSelBrandInfo : O : This is not used in the current
PackagedContent  :   : version of SET/IOTP.
BrandSelProtocol : O : This is not used in the
AmountInfo       :   : current version of SET/IOTP.
PackagedContent
BrandSelCurrency : O : This is not used in the
AmountInfo       :   : current version of SET/IOTP.
PackagedContent

```

TradingRolePackaged: O : Copied from the TradingRoleData
Content
Component. The Name Attribute of
the packaged contents must include
"Payment:" as the prefix,
for example "Payment:SET-OD".

PaySchemePackaged : M : See section 6.5.2.
Content

Output Parameters

PaySchemePackaged : M : See section 6.5.2.
Content

ContStatus : M : "Continue" must be set if there
is no problem.

7.3.3 Resume Payment Consumer

This API is used to restart a payment transaction when the transaction is suspended for some reason such as a time out. The last SET Message relevant to this suspended transaction is returned as the Response.

Input Parameters

ConsumerPayId : M : -

WalletId : O : -

PassPhrase : O : -

CallBackFunction : O : This is not used in the current version
of SET/IOTP.

CallBack : O : This is not used in the current version
LanguageList of SET/IOTP.

Output Parameters

ContStatus : M : -

PaySchemePackaged : M : See section 8.7.
Content

7.3.4 Continue Process

This API is used to pass a SET related message, received from the counter party, to the SET Bridge, and accept the next SET message as a response.

(1) Consumer Side Payment Bridge

Input Parameters

PayId	: M : Set ConsumerPayId
WalletId	: O : -
PassPhrase	: O : -
PaySchemePackaged Content	: M : See section 8.4.3.

Output Parameters

ContStatus	: M : Set "End" if SET PRes message is received in the PaySchemePackagedContent as the input parameter, otherwise set "Continue".
PaySchemePackaged Content	: O : If ContStatus is set "End", this is not used. See 8.4.3.

(2) Payment Handler Side Payment Bridge

Input Parameters

PayId	: M : Set PaymentHandlerPayId
WalletId	: O : -
PassPhrase	: O : -
PaySchemePackaged Content	: M : See section 8.4.4.

Output Parameters

ContStatus	: M : Set "End" if SET PRes message is received in the PaySchemePackagedContent as the output parameter, otherwise set "Continue".
PaySchemePackaged Content	: M : See section 8.4.4.

7.3.5. Change Process State

This API is used by the OAC to change the Process State of the OPB. For instance, it is used to change the Payment Status after a SET Payment Transaction was completed. When an error or suspend happens, this API is also used.

(1) Consumer Side Payment Bridge

Input Parameters

PayId	:	M	:	Set ConsumerPayId
ProcessState	:	M	:	-
CompletionCode	:	M	:	-
ProcessType	:	M	:	-
WalletID	:	O	:	-
PassPhrase	:	O	:	-

Output Parameters

ProcessState	:	M	:	-
CompletionCode	:	M	:	-
PercentComplete	:	O	:	See section 8.13.
xml:lang	:	O	:	-
StatusDesc	:	O	:	This field is not specified in SET/IOTP.

7.4 General Inquiry API Calls

7.4.1 Payment Instrument Inquiry

This API is not used in the current version of SET/IOTP.

7.4.2 Inquire Pending Payment

This API is used to check whether the payment Bridge or its wallet is currently in use, or not.

Input Parameters

WalletID	:	O	:	-
----------	---	---	---	---

Output Parameters

PayId	:	M	:	-
-------	---	---	---	---

7.4.3 Remove Payment Log

This API is used both Consumer and Payment Handler.

Input Parameters

PayId	:	M	:	-
WallerId	:	O	:	-
Passphrase	:	O	:	-

There is no output parameters.

7.5 Payment Related Inquiry API Calls

7.5.1 Check Payment Receipt

This API is used to check a Payment Receipt. However since the current SET specification does not support Receipts, SET/IOTP sends its own visual information of a Receipt to the SET Bridge.

Input Parameters

PayId	:	M	:	-
WalletId	:	O	:	-
PassPhrase	:	O	:	-
PaySchemePackaged Content	:	M	:	See section 8.5.1.

Output Parameters

There is no output Parameter.

7.5.2 Expand Payment Receipt

This expands an IOTP Payment Receipt Component packaged data into a form which may be used for display or printing purposes.

Input Parameters

PayId	:	M	:	-
WalletId	:	O	:	-
PassPhrase	:	O	:	-
PackagedContent	:	M	:	See section 8.5.2.

Output Parameters

BrandId	: M : -
ProtocolBrandId	: M : -
PayInstrumentId	: M : -
PaySchemePayId	: M : LID_M in the SET PRes message is set. (The format of this value must be same as SET Initiation.)
Amount	: M : Amount * AuthRatio (or CapRatio if available). CapRatio should be the high priority than AuthRatio.
CurrCodeType	: M : -
CurrCode	: M : -
PayDirection	: M : -
ProtocolId	: M : -
ProtocolTransId	: O : -
TimeStamp	: M : This value should be used the Date field of MessageWrapper in the SET PRes message
xml:lang	: O : This is not used in the SET/IOTP.
ConsumerDesc	: O : This is not used in the SET/IOTP.
PaymentHandlerDesc	: O : This is not used in the SET/IOTP.
StyleNetLocn	: O : This is not used in the SET/IOTP.
PaymentProperty	: O : This is not used in the SET/IOTP.

7.5.3 Inquire Process State

This API is used to check the payment status. For example, when the OAC receives a Continue Payment Response API, it uses this API if the ContStatus is set to "End". This API can be used at anytime.

(1) Consumer Payment Bridge

Input Parameters

PayId	: M : Set ConsumerPayId
WalletId	: O : -
PassPhrase	: O : -

Output Parameters

ProcessState : M : -
 PercentComplete : O : See 8.13 for the guideline of
 setting value.
 CompletionCode : O : See section 8.12.
 xml:lang : O : -
 StatusDesc : O : -
 PayReceiptNameRefs : O : This is not used in the SET/IOTP.
 PayReceiptPackConts: O : This is not used in the SET/IOTP.

(2) Payment Handler Payment Bridge

Input Parameters

PayId : M : Set PaymentHandlerPayId
 WalletId : O : -
 PassPhrase : O : -

Output Parameters

ProcessState : M : -
 PercentComplete : O : See section 8.13 for the guideline
 of setting value.
 CompletionCode : O : See section 8.12.
 xml:lang : O : -
 StatusDesc : O : -
 PayReceiptNameRefs : O : This is set "Pres".
 PayReceiptPackConts: O : This is not used in the SET/IOTP.

7.5.4 Start Payment Inquiry

This API call returns the SET InqReq Message in order to process a SET Inquiry.

Input Parameters

ConsumerPayId : M : -
 WalletId : O : -
 Passphrase : O : -

Output Parameters

PaySchemePackaged : M: Packaged Data to include SET
 Content InqReq message. See section 8.6.

7.5.5 Inquire Payment Status

The Payment Handler uses this API request for Consumer initiated inquiry processing. In SET/IOTP, the Payment Handler's SET Bridge receives a SET InqReq message in an InquirePaymentDetail API. The SET Core processes it, and creates a SET InqRes message. The response encapsulates the SET InqRes message.

Input Parameters

PaymentHandlerPayId: M : -
WalletID : O : -
PassPhrase : O : -
PaySchemePackaged : M : See section 8.6.
Content

Output Parameters

PaymentHandlerPayId: M : -
ProcessState : M : -
CompletionCode : O : -
xml:lang : O : -
StatusDesc : O : -
PaySchemePackaged : M : See section 8.6.
Content

8. SET dependent Process

This chapter describes the core concepts for the development of SET/IOTP.

8.1 Relationships between them for IOTP Purchase/Cash Advances

This document describes SET Initiation Messages based on the [SET EIG]. Merchant sends the 1st SET Initiation Message to the Consumer in order to activate a SET payment transaction. After this message, the other SET Initiation Messages (JPO, etc.) and the SET payment Transaction (SET PinitReq message, etc.) are exchanged between the Consumer and the Payment Handler.

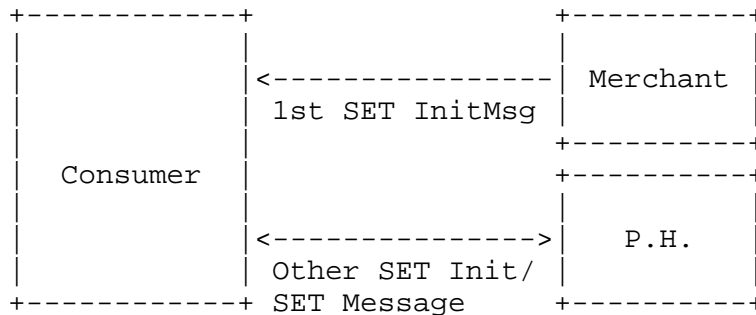


Figure 11 Relationship between IOTP Messages and SET Messages

When the Merchant sends any data (e.g., SET SaleDetail) except a SET Related messages (e.g., SET PinitRes message), it can send it by two different methods:

- (a) The Merchant sends the data via the Consumer.
- (b) The Merchant sends the data out-of-band.

In case (a), the Merchant sends the data by encapsulating it into `TradingRoleData.PackagedContent` inside the Offer Response Block sent to Consumer. The data is copied to the Payment Request Block and sent to the Payment Handler. This case assumes that the format of the data is already agreed upon between the Merchant and the Payment Handler.

This document does not specify case (b).

8.2 Definition of Identifiers

8.2.1 Definition of BrandId

BrandId should be used registered identification for IANA. Now, the following BrandIds have registered:

Amex, Dankort, JCB, Maestro, MasterCard, MICOS, VISA, atCredits, EZpay, GeldKarte, Mondex, paybox

8.2.2 Definition of ProtocolBrandId

ProtocolBrandID is defined as follows:

<Premise> SET BrandID is defined as `brand[:Product]`. ([:] is indicated as optional.) In SET, The brandID is a brand name, which corresponds to the brand of the payment card. Additionally the Product is a product name, which is defined as the type of product within the specific brand such as Gold Card.

Set IOTP ProtocolBrandId as follows:

brand:Product:PCN

In here,

- o The brand above is the same as the sub data of SET BrandID, as Brand Name (brand), defined in SET.
- o Product above is the same as the sub data of SET BrandID, as Product Name (Product), defined in SET.
- o PCN above is the Promotional Card Name, and is written in the SET Certificates.

Example:

Visa:Gold:WalMart

Since SET Brand ID has a colon between brand and Product, the two colons should be able to delimit Brand, Product, and PCN.

Product and PCN can omit if necessary. For the detail of these definitions are follows:

(1) The case of omitting Product

Definition: brand::PCN

Example: VISA::UC_VISA

(2) The case of omitting PCN

Definition: brand:Product

Example: VISA:Gold

(3) The case of omitting Product, PCN

Definition: brand

Example: VISA

Invalid Examples:

VISA:Gold:

VISA::

VISA:

ProtocolBrandId which there is no brand.

8.2.3 Definition of ProtocolId

ProtocolId defines as follows:

ProtocolId := SETName + Version

SETName := "SET"

Version := "v" + version + "." + revision

Where the version is number matching a major SET version, and the revision is the number matching a minor SET revision.

Example:

"SETv1.0", "SETv2.0"

NOTE: In the current version of SET/IOTP, "SETv1.0" is fixed as ProtocolId.

8.2.4 Relationship between Ids

ProtocolBrandId must be unique and depends on BrandId and ProtocolId. The followings are map among BrandId and ProtocolId, which have registered in IANA, and ProtocolBrandId.

BrandId	ProtocolId	ProtocolBrandId

Amex	SETv1.0	Amex
Dankort	SETv1.0	Dankort
JCB	SETv1.0	JCB
MasterCard	SETv1.0	MasterCard
Nicos	SETv1.0	Amex
VISA	SETv1.0	VISA

Regarding to the BrandIds except above, the BrandId registrant (e.g., credit card company) MUST register it in order to be able to map one to one between ProtocolBrandId and the pair of BrandId and ProtocolId.

8.3 Process prior to Payment

8.3.1 FindAcceptedPaymentProtocol Function

(1) Parameter of PayProtocolPackagedContent

Name : O : This is not used in SET/IOTP.
 Content : M : This should be set "PCDATA".
 Transform : M : This is set "BASE64".
 ContentData : M : SET specific protocol data. Includes data that is used to create the 1st SET Initiation Message that is not contained in other IOTP elements.

(2) Parameter in the ContentData

Parameters of ContentData are described below. The Field Values follow the [SET EIG].

Field	Required
-----	-----
MIME-Version	Optional
Content-Transfer-Encoding	Mandatory
SET-Initiation-Type	Mandatory
SET-LID-M	Optional
SET-InstallTotalTrance	Optional
SET-Recurring	Optional
SET-Ext-OID	Optional
SET-Ext-Data	Optional
SET-Ext-Mandatory	Optional
SET-Echo-In-Response	Optional
SET-Echo-In-Request	Optional

For Example:

MIME-Version: 1.0
 Content-Transfer-Encoding: Binary
 SET-Initiation-Type: Payment-Initiation
 SET-Recurring: 31 19960223
 SET-Service-URL: http://www.custcare.com/index.html
 SET-LID-M: 515A533033363632594B

Note: The contents in ProtocolPackagedContent must be US-ASCII and encoded by BASE64.

8.3.2 FindPaymentInstrument Function

(1) Information of PayInstrument

Returns a list of Payment Instrument IDs related to the BrandId and ProtocolBrandId. In this document, BrandId and ProtocolId are defined in section 8.2.

In this document, Brand has two recognized meanings in SET/IOTP, as follows:

Brand as Primary Brand:

The Primary Brand is the Brand which is defined as brand in SET, such as VISA, MasterCard, Nicos.

Brand as Dual Brand or Promotional Brand:

The Dual Brand is the payment instrument which has two Brand, such as UC-VISA (UC Card and VISA Card) This style is popular in Japan.

A Promotional Brand means that, if the Consumer pays with that Brand, then the Consumer will receive some additional benefit such as discount or frequent flyer point.

1. ProtocolBrandId as a Primary Brand

Example:

"MasterCard", "MasterCard::UC", "MasterCard:Gold:" and "MasterCard::WalMart" are all MasterCard Brands.

2. ProtocolBrandId as a Dual Brand or a Promotional Brand

Example:

"MasterCard::UC" is Dual Brand of "MasterCard" and "UC".
"SET:MasterCard::WalMart" is Promotional Brand of MasterCard-WallMart.

The SET Bridge receives the ProtocolBrandId from the OAC in the FindPaymentInstrument Function,

(1) If the accepted ProtocolBrandId is XXX:YYY

The SET Related Module searches for ProtocolBrandIds with the string "XXX:YYY:*" (* is wild card), the corresponding PaymentInstrumentIds of all ProtocolBrandIds with the matching Primary Brand (regardless of also being a Dual Brand or Promotional Brand) will be returned to the OAC, for the Consumer to select from.

(2) If the accepted ProtocolBrandId is XXX:YYY:ZZZ

The SET Related Module searches for ProtocolBrandIds with the string "XXX:YYY:ZZZ", only the corresponding PaymentInstrumentIds of the ProtocolBrandIds that match the Dual Brand or Promotional Brand will be returned to OAC, for the Consumer to select from.

Example:

Assume ProtocolBrandIds are correspond to PaymentInstrumentIds in the SET Bridge as follows,

ProtocolBrandId	PaymentInstrumentId

MasterCard	1
MasterCard::UC	2
MasterCard::WallMart	3
VISA::UC	4

If the SET Bridge receives a ProtocolBrandId as "MasterCard" in the FindPaymentInstrument Function, the SET Bridge will return "1", "2", and "3". However, if the SET Bridge receives a ProtocolBrandId as "MasterCard::UC" to OAC, SET Bridge will returns only "2".

8.3.3 GetPaymentInitializationData Function

(1) Create TradingRolePackagedContent

If necessary, The SET Related Module generates TradingRolePackagedContent corresponded to the received ReceiverOrgID. The ContentData of TradingRolePackagedContent is the information which the Payment Handler needs to process the SET Transaction (for example, the SET SaleDetail. and the SET OD). The ContentData, Content, and the Transform must be agreed upon between the Merchant and the Payment Handler beforehand.

The Name Attribute of the packaged contents must include "Payment:" as the prefix, for example "Payment:SET-OD". If there is no PackagedContent corresponding to ReceiverOrgID, such that the SET Related Module does not need to create the PackagedContent, the TradingRolePackagedContent is not created.

Parameters in TradingRolePackagedContent

```

-----
Name          : O : This is not specified in the current SET/IOTP.
Content       : M : Should be identical between the Payment
                  Handler and the Merchant.
Transform     : M : Should be identical between the Payment
                  Handler and the Merchant.
ContentData   : M : Element Data for the Payment Handler to
                  process the SET Transaction. Should be
                  identical between the Payment Handler and
                  the Merchant.

```

8.4 Process of Payment

8.4.1 StartPaymentConsumer Function

(1) Process of the 1st SET Initiation Message

Since there are similar items between the SET Initiation Message Fields and IOTP Elements, IOTP elements can be used for the corresponding SET Initiation Fields. Other SET Initiation Fields, except URL information (for detail, see below), is encapsulated in the PayProtocolPackagedContent.

This document does not specify how the SET Related Module implements the 1st SET Initiation Process.

The following table shows the list of SET Initiation Fields that corresponds to IOTP Elements.

SET Initiation Field	IOTP Element (in TPO.Brandlist)
SET-Version	Consumer selected ProtocolId
SET-Brand	Consumer selected ProtocolBrandId
SET-Amount	Consumer selected Amount Data in CurrencyAmount.
SET Initiation Field	IOTP Element (in OfferResp)
Order Description	The hash data of ContentData of PackagedContent of Order Component.

(b) SET-Version:

SET-Version can be corresponded to ProtocolId. The version number appears after the "v" for the SET-Version.

```
ProtocolId -> _____
               SETv1.0
               ~~~<- SET-Version
```

Figure 12 ProtocolId vs SET-Version

(c) SET-Brand:

SET-Brand can be corresponded to ProtocolBrandId.

(d) SET-PurchAmt:

It is necessary to adjust the format of the Amount between IOTP and SET, since IOTP and SET use different syntax.

Assumption:

- o In SET/IOTP, The "ISO4217-A" (the currency code which is represented by three alphabet, such as "USD") is mandatory.
- o Consumer Side SET Related Module should have a mapping table between "ISO4217-A" and "ISO4217-N" (the currency code which is represented by three digit, such as "840").

(d) -1 Content of the SET-PurchAmt

The content of the SET-PurchAmt is as follows:

SET PurchAmt: currency amount amtExp10

For a description see [SET] Book 2, page 299. For example, \$129.50 is represented by "840 12950 -2". In this case, the corresponding values for the "currency", "amount" and "amtExp10" are "840", "12950" and "-2" respectively.

(d) -2 Content of IOTP Amount Elements

The content of the three IOTP amount elements consist of the following: Amount, CurrCodeType and CurrCode. For a description of each, see [RFC 2801]. For example, \$129.50 is represented by the following:

```
CurrCodeType="ISO4217-A"
CurrCode="USD"
```

Amount="129.50"

(d) -3 Example of how-to-translate

The one-to-one mapping between the IOTP format and the SET format is very simple. This example of sequence below uses the example of IOTP amount Element above.

- 1) Translate from IOTP CurrCode (ISO4217-A) to SET currency (ISO-4217-N). For example, if CurrCode="USD", then the value of currency is "840".
- 2) Calculate how many decimal places are represented in the Amount. For example, if Amount="129.50", there are "2" decimal places.
- 3) [The number of decimal places] * (-1) corresponds to the SET amtExp10. In the above case, SET amtExp10 = 2 * (-1) = -2.
- 4) $10^{\text{[The number of the Amount's decimal places]}} * \text{Amount}$ corresponds to the SET amount. In the above example, SET amount = $10^2 * 129.50 = 12950$.
- 5) Concatenate three integers and use white spaces as a delimiter.

Finally, in the above case, the SET PurchAmt is represented as "840 12950 -2".

(e) SET OD (Order Description) vs. IOTP Order Information

In the IOTP, the OAC handles the Order Information, such as display use, as SET uses the Order Information. Payment Handler does not know the actual Order Information because the Merchant and Payment Handler may exist in the separate domains. However, Payment Handler needs to get the SET OD from Merchant via the Consumer or directly because Payment Handler needs the SET OD to create 2nd SET Initiation message and after. In this situation, the Merchant should not pass the actual order information to the Payment Handler because the order information may be considered private data. Therefore, SET/IOTP defines SET OD as the hash of IOTP Order Information. The hash algorithm must be SHA1.

But the Order Component may be included two or more Packaged Content (see [RFC 2801]). Therefore SET/IOTP specifies to create hash as follows:

(e) -1. If the Name attribute does not have the Name attribute, such that the Order Component have only one Packaged Content, hash the Contents Data using SHA1 simply and be encoded by BASE64.

(e) -2. Otherwise, such that there exists the Name attribute, sort the Packaged Contents in the UTF-16 character code order of Name attribute and hash the Content Data using SHA1 and concatenate them in proper sequence, then hash it using SHA1 again and be encoded by BASE64.

NOTE:

To avoid different character encodings between applications, in this document, SET OD MUST be constructed from the ContentData in OrderPackagedContent as follows:

- (1) Convert it to network byte ordered Unicode encoding data.
- (2) Hash (1) using SHA1
- (3) Convert (2) to BASE64 US-ASCII data

Therefore, "Content-Type","charset" MUST be "text/plain","us-ascii" respectively when SET Initiation message is constructed.

- (f) SET-***-URL vs. IOTP Net Location

In IOTP, the OAC handles location data therefore the OAC does not need to pass net location data on to the OPB. However, some vender implemented consumer SET/IOTP wallets may need the URL information to process the SET Initiation. Thus, if necessary, the Consumer's SET Related Module must set appropriate URL data to SET-***-URL.

- (2) Create the next SET related message

Generate SET related message (SET PInitReq or SET Initiation Response) at the SET Related Module, to be sent to the Payment Handler.

- (3) Error check of the next SET related message.

If SET related message which is created in (2) is SET Initiation Response and includes any error in it, SET Related Module creates an ErrorResponse message with ErrorCode to "EncapProtErr" and the Severity to "HardError" and sent it to the OAC.

(4) Create PaySchemePackagedContent

The followings are the parameter of PaySchemePackagedContent in StartPaymentConsumerResponse.

ContentData : M : SET Related Message which is encoded by
BASE64. (e.g., SET PinitRes message or
SET Initiation Response Message)
Name : O : This is not used in the current SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".

(5) Store of the Payment Information

SET Bridge should store the following in the DataBase:

- o ConsumerPayId
- o PaySchemePackagedContent
- o ContStatus
- o ContentSoftwareId (corresponding to the PaySchemePackagedContent)
- o ProcessState

8.4.2 StartPaymentPaymentHandler Function

(1) Process for TradingRoleData

SET Bridge must processes appropriately, for example pass it to the SET Core, if there exists the TradingRolePackagedContent as the input Parameter.

(2) SET Specific Process

The SET Related Module processes the SET Initiation Response or the SET Transaction (SET PInitReq). In addition, the SET Related Module generates a message (the next SET Initiation Message or SET PInitRes) corresponding to the results of the processed message. This message will be sent to the Consumer.

(3) Error check of the next SET related message.

If SET related message which is created in (2) includes any error, SET Related Module create an ErrorResponse message with ErrorCode to "EncapProtErr" and the Severity to "HardError" and sent it to the OAC.

(4) Generate PaySchemePackagedContent

PaySchemePackagedContent which Encapsulate the SET Initiation Message or SET PInitRes into ContentData and generate the PaySchemePackagedContent. The Parameters of PaySchemePackagedContent as Output is as follows:

```
ContentData : M : SET Related Message which is encoded by
                  BASE64 (e.g., SET PinitRes message or
                  SET Initiation Response Message).
Name         : O : This is not used in the current SET/IOTP.
Content      : M : This field should be set to "PCDATA".
Transform    : M : This must be set "BASE64"
```

8.4.3 ContinueProcess Function (Consumer Side)

(1) SET Specific Process

The Parameters of PaySchemePackagedContent as Input is as follows:

```
ContentData : M : SET Related Message which is encoded by
                  BASE64 (e.g., SET PinitRes message,
                  SET PRes message or SET Initiation
                  Response Message).
Name         : O : This is not used in the current SET/IOTP.
Content      : M : This field should be set to "PCDATA".
Transform    : M : This must be set "BASE64"
```

SET Related Module processes the SET Related Message in the PaySchemePackagedContent, then SET Related Message corresponding to the processed message is created if necessary.

(2) SET Related Message Error Check

If SET related message which is created in (2) includes any error, SET Related Module create an ErrorResponse message with ErrorCode to "EncapProtErr" and the Severity to "HardError" and sent it to the OAC.

(3) Create PaySchemePackagedContent

The followings are the parameter of PaySchemePackagedContent in ContinueProcessResponse.

```
ContentData : M : SET Related Message which is encoded by
                  BASE64 (e.g., SET PinitReq message,
                  SET PReq message or SET Initiation
                  Response Message).
```

Name : O : This is not used in the current SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".

If the ContentData which has received from Payment Handler is SET Pres message, this data is not created.

8.4.4 ContinueProcess Function (Payment Handler Side)

(1) Brand Integrity Check between IOTP Elements and SET Elements

Since the Consumer sets the Amount and Brand in the SET Message, based on the IOTP message, it might be altered when the IOTP message is copied to the SET message. Thus, the Payment Handler needs to check the Elements in IOTP components (Payment, etc.) and the Elements in the SET message to make sure they are consistent. The IOTP Brand specified by the Merchant should correspond to the Brand used in the SET payment.

The Brand Integrity check sequence is as follows:

(a) After receiving the SET PReq message, check the Consumer selected Brand information (e.g., ProtocolBrandId) in the IOTP Payment Request against information in the SET certificate in the SET PReq message.

(b) If they do not match, return a SET Bridge Level Error (Severity="HardError", ErrorCode="AttNotValid" and Names="BrandId").

Additionally, the SET PReq message signature must be verified with the SET CardHolder's certificate. (This is done during a normal SET Transaction.)

NOTE: This integrity check is necessary even if There is no Promotional Card Name in the ProtocolBrandId because SET may have selected the MasterCard even though IOTP has selected the VISA.

(2) SET Related Process

Encapsulate the SET related Message (SET Initiation Message or SET Transaction Message) in to Content Data of PaySchemePackagedContent and send it to the Sender.

The followings are the parameters of PaySchemePackagedContent as output.

ContentData : M : SET Related Message which is encoded by
BASE64 (e.g., SET PinitReq message,
SET PReq message or SET Initiation
Response Message).
Name : O : This is not used in the current SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".

(3) SET Related Message Error Check

If SET related message which is created in (2) includes any error, SET Related Module create an ErrorResponse message with ErrorCode to "EncapProtErr" and the Severity to "HardError" and sent it to the OAC.

If SET related message which is created in (2) is SET Pres message, and its message includes except:

(a) CompletionCode in SET Pres message is "authorizationPerformed" and AuthCode is "Approved" or (b) CompletionCode in SET Pres message is "capturePerformed" and CapCode "Success",

SET Related Module create ErrorResponse message with ErrorCode to "BusinessError" and the Severity to "HardError" and sent it to the OAC.

(4) Create PaySchemePackagedContent

The followings are the parameter of PaySchemePackagedContent in ContinueProcessResponse.

ContentData : M : SET Related Message which is encoded by
BASE64 (e.g., SET PinitRes message,
SET Pres message or next SET Initiation
Message).
Name : O : "Pres" only if ContentData includes
SET Pres message, otherwise this is
not used in the current SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".

If ContentData includes the SET Pres message, ContStatus MUST be "End".

8.4.5 InquireProcessState Function

(1) Setting ProcessState

Values for the ProcessState are described in section 8.9.2.

(2) Setting CompletionCode

Set to "Unspecified" when a SET Business Failure has occurred, and set StatusDesc to the value corresponding to AuthCode or CapCode.

(3) Setting StatusDesc

The values for PayStatusDesc are not specified in the SET/IOTP.

(4) Create PayReceiptNameRefs

Set to "Pres" in the PayReceiptNameRefs

8.5 Payment Receipt

8.5.1 CheckPayReceipt Function

SET Related Module does not check the Payment Receipt Information especially, sends the general response message as long as valid request message.

The Parameters of PayReceiptPackagedContent are followings:

Name	:	O	:	This MUST be set "Pres"
Content	:	M	:	This field should be set to "PCDATA".
Transform	:	M	:	This must be set "BASE64".
ContentData	:	M	:	SET Pres message which is encoded by BASE64.

8.5.2 ExpandPayReceipt Function

(1) PayReceiptPackagedContents

The Parameters of PayReceiptPackagedContent are as follows:

Name	:	O	:	This MUST be set "Pres"
Content	:	M	:	This field should be set to "PCDATA".
Transform	:	M	:	This must be set "BASE64".
ContentData	:	M	:	SET Pres message which is encoded by BASE64.

(2) Get the current status information

SET Related Module gets out the following element from Data Base using ConsumerPayId, PaymentHandlerPayId as keys.

- o BrandId
- o ProtocolBrandId
- o PayInstrumentId
- o Amount
- o CurrCodeType
- o CurrCode
- o PayDirection

(3) Get the SET Data

SET Related Module gets the following data from SET Pres message which take as the Request Message.

(a) Date Field in the MessageWrapper Date field between SET and IOTP is slightly different. The different things are as follows:

- o There is no TimeZone in the Date field of SET.
- o Second and Milli-second can be omitted in the Date field of SET

Therefore, SET Related Module needs to compensate the Date information when TimeStamp field is set.

(b) AuthRatio in SET Pres message. (CapRatio is high priority than AuthRatio if available.)

(c) LID_M in SET Pres message. (The style of this value is the same as it of SET Initiation message.)

8.6 Status Inquiry

In SET/IOTP, SET Inquiry Initiation is not supported (i.e., omitted). SET Inquiry Messages are embedded in the PaySchemeData element in IOTP Inquiry Messages.

The Parameters of PaySchemePackagedContent in StartPaymentInquiryResponse are follows:

Name	: O	: This is not used in the SET/IOTP.
Content	: M	: This field should be set to "PCDATA".
Transform	: M	: This must be set "BASE64".
ContentData	: M	: SET InqReq message which is encoded by BASE64.

The Parameters of PaySchemePackagedContent in InquirePaymentStatus are follows:

Name : O : This is not used in the SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".
ContentData : M : SET InqReq message which is encoded by BASE64.

The Parameters of PaySchemePackagedContent in InquirePaymentStatusResponse are follows:

Name : O : This is not used in the SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".
ContentData : M : SET InqRes message which is encoded by BASE64.

The Parameter of PaySchemePackagedContent in ContinueProcess are follows:

Name : O : This is not used in the SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".
ContentData : M : SET InqRes message which is encoded by BASE64.

8.7 Resume Process

The Parameter of PaySchemePackagedContent in RequePaymentConsumerResponse are as follows:

Name : O : This is not used in the SET/IOTP.
Content : M : This field should be set to "PCDATA".
Transform : M : This must be set "BASE64".
ContentData : M : SET Related Message which is encoded by
BASE64 (e.g., SET PinitRes message
or SET Initiation Response Message).

8.8 SET Scheme Specific Authentication on IOTP

IOTP authentication, which uses the SET Scheme, is not used in SET/IOTP.

8.9 SET Bridge ProcessState

8.9.1 SET Bridge ProcessState of Consumer

No Status ----> InProgress	: When StartPaymentConsumer Function is called
InProgress ---> InProgress	: When ContinueProcess Function is called : When ChangeProcessState Function (ProcessState="Failed") is called
InProgress ---> ProcessError	: When ChangeProcessState Function (ProcessState="ProcessError") is called : The Technical Error (Hard Error) is occurred in SET Bridge
InProgress ---> CompletedOK	: When ChangeProcessState Function (ProcessState="CompletedOK") is called
InProgress ---> Failed	: When ChangeProcessState Function (ProcessState="failed") is called : The Business Error is occurred in SET Bridge
InProgress ---> Suspended	: When ChangeProcessState Function (ProcessState="Suspended") is called : ErrorCode="ResumeRequired" is is occurred.
Suspend ---> InProgress	: ResumePaymentConsumer Function is called
Suspend ---> ProcessError	: When ChangeProcessState Function (ProcessState="ProcessError") is called (the Technical Error is occurred prior to ResumePayment-Consumer Function call) : The Technical Error (Hard Error) is occurred in SET Bridge (the Technical Error is occurred while ResumePaymentConsumer is calling)

8.9.2 SET Bridge ProcessState of Payment Handler

No Status ----> InProgress	: When StartPaymentPaymentHandler is called
InProgress ---> InProgress	: When ContinueProcess Function is called : When ChangeProcessState Function (ProcessState="Failed") is called
InProgress ---> ProcessError	: When ChangeProcessState Function (ProcessState="ProcessError") is called : The Technical Error (Hard Error) is occurred in SET Bridge : SET Error Message is occurred
InProgress ---> CompletedOK	: When SET Transaction is completed.
InProgress ---> Failed	: When ChangeProcessState Function (ProcessState="failed") is called : The Business Error is occurred in SET Bridge
CompletedOK ---> Failed	: When ChangeProcessState Function or CancelPayment Function (ProcessState="Failed") is called and the payment is cancelled.

8.10 Relationship between Pay Step and Deliv Step on SET/IOTP

SET/IOTP recommends the following regarding Delivery:

Physical Goods

For physical goods, the IOTP Delivery Exchanges should be omitted. That is, set DelivExch=False and DelivAndPayResp=False in the Delivery Component. This is to avoid the situation where the IOTP Delivery Handler must check with the IOTP Payment Handler on the status of a credit authorization. When a Delivery Inquiry transaction might occur, the DelivReqNetLocn attribute in the DeliveryData Element must have been specified at the time of the original Offer Response Message. If you want to use the Delivery Exchange, you need to process the inquiry of the credit authorization out of IOTP between IOTP Payment Handler and Delivery Handler.

Digital Goods

 For digital goods sold through SET/IOTP, authorization should be processed on a real-time basis.

8.11 Completion Code

In SET/IOTP, the CompletionCode, which is a Business Error Code, is set as follows:

Value	Description
BrandNotSupp	This value is not used.
CurrNotSupp	This value is not used.
AuthError	The IOTP Authentication has failed for any reason.
InsuffFunds	This value is not used.
InstBrandInvalid	This value is not used.
PaymentDecl	A SET business failure has occurred.
InstNotValid	This value is not used.
BadInstrument	This value is not used.
Unspecified	Unspecified error. There is some known problem or error, which does not fall into one of the other CompletionCodes.

8.12 PercentComplete

This document recommends to set the PercentComplete as follows:

SET Related Message	Setting for Consumer	Setting for Paymnet Handler	Value of PercentComplete
SET Initia- tion	After 1st SET Initiation Response has Cteated (See Note)	After 1st SET Initiation Response has Processed (See Note)	20
SET PinitReq	After Created	After Processed	40
SET PinitRes	After Processed	After Created	60
SET PReq	After Created	After Processed	80
SET PRes	After Processed	After Created	100

Note: According to the SET Initiation, PercentComplete should be set "20" at the timing of 1st SET Initiation Response is created/processed because number of its message is variable.

8.13 Severity

In the current version of SET/IOTP, if a technical error occurs in the SET Bridge, the Severity has to be always set to "HardError".

9. Error Handling

This chapter describes types of handling Errors.

9.1 Types of Errors

SET/IOTP defines the following error types:

(1) IOTP Level Error

This is defined as an error which is NOT specified in [SET EIG] nor [SET]. IOTP Level Errors are divided into two types according to the following:

OAC Level Error: Error in the OAC. This error is defined in the [IOTP].

SET Related Module Level Error: Error generated in by process on the SET Related Module, not specified in [SET EIG] nor [SET]. For example, when checking the consistency between SET and IOTP elements on SET Related Module, an error might be returned to OAC.

(2) SET Level Error

This is defined as an error which is specified in [SET EIG] or [SET]. SET Level Errors have been divided into two types of error according to following:

SET Technical Level Error: Error in the SET Related Module. This error is defined in [SET] or [SET EIG]. SET Technical Level Errors are further subdivided into two types of errors:

(a) SET Initiation Error Error while the SET Initiation Process is in progress.

(b) SET Transaction Error Error when the SET Transaction (SET PInitReq message, SET PReq message, etc.) is in progress.

SET Business Level Error: Error when a business error (e.g., an authorization failure) occurs while the SET Transaction is being processed. In SET, Business Level Errors will be returned in the SET Pres message. SET does not use a SET Error Message for this type of error. However, it is necessary to present the OAC with what kind of SET Business Error has occurred.

In this below, the details of each errors above are described.

9.2 IOTP Level Error (OAC Error)

When OAC Level Errors have occurred, if necessary, the sender and receiver must issue ChangeProcessState API and change the status. For the detail of these errors, see [IOTP].

9.3 IOTP Level Error (SET Bridge Error)

This is the error generated in a process on the SET Related Module, not specified in [SET EIG] nor [SET]. For example, when checking the inconsistency between SET and IOTP elements on SET Related Module, it might cause an error. This error should be notified to OAC.

In this case, as a response message, Payment Scheme Data is not returned. An appropriate information must be set to Status Response.

9.4 SET Level Error (SET Technical Error)

9.4.1 SET Initiation Error

There are two SET Initiation errors as follows:

- o Error generated in SET Initiation Message
- o Error generated in SET Initiation Response Message.

(1) SET Initiation Message Error

[SET EIG] describes the error handling when a problem rises in SET Initiation Message. So the Consumer will do the same error handling in 9.4.2.

When SET Initiation Error rises in 1st Initiation Message, an error message will be returned to the Merchant. If an error occurs after 2nd Initiation Message, an error message will be returned to the Payment Handler. SET Initiation Response will be generated having SET-Error-Field in Response Message Header and will be returned ErrorCode as "PayEncapError" and Severity as "HardError".

(a) SET Initiation Response Error

In SET EIG, there is no description about the handling on the problems in SET Initiation Response. However, it is necessary to define some handling for the problems in SET/IOTP

(b) Process of Payment Handler

When a problem rises in SET Initiation Response, SET Related Module generates ErrorResponse, which is included the "EnCapProtoErr" as ErrorCode and the "HardError" as Severity. But PaySchemePackagedContent is not included in this API.

(2) Process of Consumer

ChangeProcessState API must be issued, and ProcessState must be modified.

9.4.2 SET Transaction Error

(1) Process of Sender

When a SET Transaction Error rises, SET Core creates SET Error Message. Then the SET Related Module creates ErrorResponse Message which includes "HardError" as Severity, "EnCapProtoErr" as ErrorCode and PaySchemePackagedContent. The SET Bridge passes the ErrorResponse Message to OAC. OAC will generate an Error Block which includes PaySchemePackagedContent and sends it to the Receiver side.

(2) Process of Receiver

With ContinueProcess API, receiver's OAC sends the message including the PaySchemeData to SET Bridge. SET Bridge passes the SET Error Message to SET Core for this process. After that, SET Bridge sends "End" status with ContinueProcessResponse API.

9.5 SET Level Error (SET Business Error)

(1) Process of Payment Handler

SET Related Module checks the SET Business Error in StatusCode in SET Pres message. When SET Transaction Error occurs, SET Related Module creates ErrorResponse Message which is included SET Pres as PaySchemePackagedContent and ErrorCode as "BusinessError" and returns it to OAC. OAC creates Payment Response Block after gets the SET scheme specific receipt in InquireProcessState/Response, and sends it to the Consumer.

(2) Process of Consumer

SET Related Module conducts the same process as in the process that Consumer receives Payment Response Block.

10. Security Considerations

In the IOTP, Merchant and Payment Handler may exist in different domains. So, if the Merchant passes the payment related information to the Payment Handler via the Consumer, the payment security level may depend on the IOTP. If you want to avoid this, you will need to check integrity of these data by using out-of-band communication between the Merchant and the Payment Handler. In this case, the security level depends on the communication path between them.

11. References

The following books provide essential background material. Readers are strongly encouraged to consult these references for more information.

- [BASE64] Base64 Content-Transfer-Encoding. A method of transporting binary data defined by MIME. See: RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. N. Freed & N.Borenstein. November 1996.
- [RFC 2801] Burdett, D., "Internet Open Trading Protocol - IOTP, Version 1.0", RFC 2081, April 2000.
- [SET] SET Secure Electronic Transaction (TM) , Version 1.0, May 31, 1997
 Book 1: Business Description
 Book 2: Programmer's Guide
 Book 3: Formal Protocol Definition
- [SET EIG] External Interface Guide to SET Secure Electronic Transaction, Sep 24, 1997.
- [SJR] "SET Secure Electronic Transaction Specification" Support for Japanese Requirements, Mar 16, 1998.
- [IOTP Payment API] Hans, W., et al., "Payment API for v1.0 Internet Open Trading Protocol (IOTP)", Work in Progress.

- [ISO4217] ISO 4217: Codes for the Representation of Currencies. Available from ANSI or ISO.
- [XML] Extensible Mark Up Language. A W3C recommendation. See <http://www.w3.org/TR/1998/REC-xml-19980210> for the 10 February 1998 version.

12. IANA Considerations

This document does not ask for any action from IANA. It references an existing registry, `iotp-codes`, where at the time of publication of this RFC the following BrandID's are registered:

Amex, Dankort, JCB, Maestro, MasterCard, MICOS, VISA, atCredits, EZpay, GeldKarte, Mondex, paybox

13. Acknowledgement

The author of this document appreciates the following contributors to this protocol (in alphabetic order of company) without which it could not have been developed.

Andrew Drapp	Hitachi Europe, Ltd.
David Burdett	Commerce One (ex. Mondex International)
Donald Eastlake 3rd	Motorola (ex. IBM)
Hans-Bernhard Beykirch	SIZ
John Wankmuller	MasterCard International
Mark Linehan	IBM
Richad D. Brown	Kedemon (ex. Globe SET)
Werner Hans	SIZ

14. Author's Address

Yoshiaki Kawatsura
Hitachi, Ltd.
890 Kashimada Saiwai-ku Kawasaki-shi
Kanagawa, 212-8567 Japan

EMail: kawatura@bisd.hitachi.co.jp

15. Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

